

**АРКАБАЕВ Н.К.
КОШМУРЗАЕВ И.Б.**

РЕШЕНИЕ

**ОЛИМПИАДНЫХ
ЗАДАЧ ПО
ИНФОРМАТИКЕ**



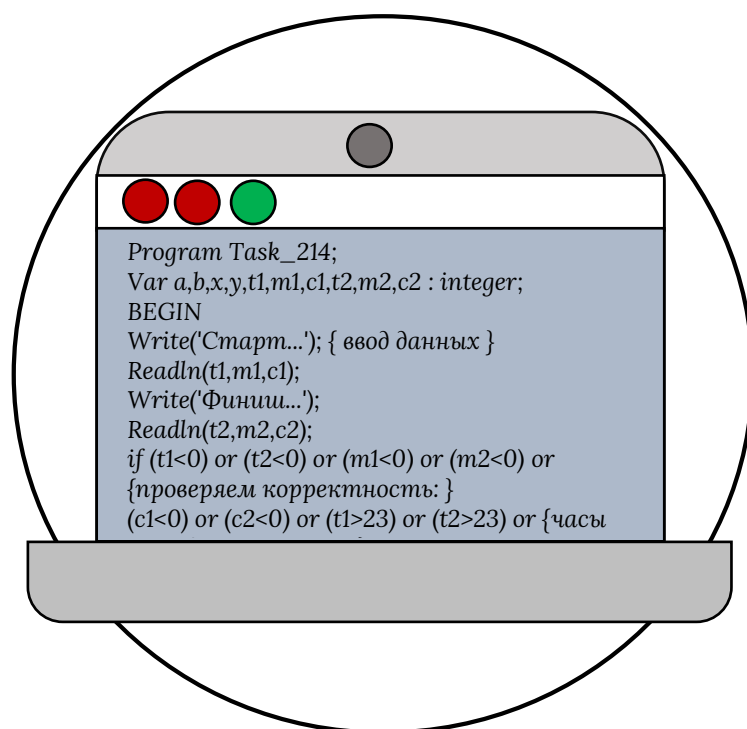
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И
НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

ОШСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

АРКАБАЕВ Н.К.
КОШМУРЗАЕВ И.Б.

РЕШЕНИЕ

ОЛИМПИАДНЫХ ЗАДАЧ ПО ИНФОРМАТИКЕ



г. Ош 2019

УДК 002
ББК 73 873
А – 82

Рецензенты:

Арапбаев Р.Н. – кандидат физико-математических наук,
доцент.

Мурзакулова Т.А. – Заведующий кабинета информатики
Ошского института образования.

А – 82 Аркабаев Н.К. Кошмурзаев И.Б.
Решение олимпиадных задач по
информатике. – Ош: 2019. – 176 с.

ISBN 978-9967-18-528-9

Этот сборник содержит подробные решения 130 олимпиадных задач, данных в рамках олимпиады по предмету информатика и программирования. Он может быть интересен школьникам, студентам и всем, кто хочет участвовать в олимпиады проводимых по информатике в школах и по программированию в ВУЗах.

В качестве дополнительного материала прилагаются тексты решений всех задач для сред Free Pascal, PascalABC .Net и подробное описание решения задачи.

**Сборник «Решение олимпиадных задач по информатике»
рекомендован к публикации решением Ученого Совета
ОшГУ. Протокол №227 от 25.02.2019 г.**

A140400000019
ISBN 978-9967-18-528-9

УДК 002
ББК 73 873

**© Аркабаев Н.К.,
Кошмурзаев И.Б.**

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ	7
Задача №001. A в степени N	8
Задача №002. Число действий порядка $\log_2 n$	8
Задача №003. Вычисления факториалу	9
Задача №004. Вычисления суммы факториалов	10
Задача №005. Вычисление НОД	11
Задача №006. НОД и уравнения	11
Задача №007. Десятичные знаки числа	12
Задача №008. Функция	13
Задача №009. Абракадабра	14
Задача №010. Количество различных чисел	16
Задача №011. Протокол	16
Задача №012. Длинная арифметика	18
Задача №013. Счастливые билеты	19
Задача №014. Поиск палиндромов	21
Задача №015. Курьеры	22
Задача №016. Треугольник	23
Задача №017. Попал-не-попал	24
Задача №018. Произведение	25
Задача №019. Поиск различных цифр	26
Задача №020. Сумма 17-ти	27
Задача №021. Уравнение	28
Задача №022. Среднее арифметическое	29
Задача №023. Макси элемент	30
Задача №024. Первый ноль	31
Задача №025. Дальний минус	32
Задача №026. Больше семи	33
Задача №027. Делители	34
Задача №028. Простые делители	35
Задача №029. Замена	36
Задача №030. Таблица	37
Задача №031. Сумма ряда №1	38
Задача №032. Сумма ряда №2	38

Задача №033. Выборы	40
Задача №034. Деление на 13 или 17	41
Задача №035. Муравей.....	42
Задача №036. Подсчет чисел.....	45
Задача №037. Три числа	47
Задача №038. Поиск чисел №1.....	47
Задача №039. Поиск чисел №2	48
Задача №040. Самая длинная цепочка.....	49
Задача №041. Сообщество роботов.....	51
Задача №042. Физическая лаборатория	53
Задача №043. Счастливые номера	56
Задача №044. Перевозка грузов	57
Задача №045. Знак #.....	58
Задача №046. Произведение в строке	60
Задача №047. Замена слова местами	61
Задача №048. Площадь многоугольника.....	63
Задача №049. Окружности	65
Задача №050. Делители	65
Задача №051. Дробь	66
Задача №052. Большие числа.....	68
Задача №053. Игра.....	69
Задача №054. Черно-белый рисунок	70
Задача №055. Сумма больших чисел	71
Задача №056. Алфавит	72
Задача №057. Пары.....	73
Задача №058. Номер страницы	73
Задача №059. Шахмат.....	75
Задача №060. Призма.....	78
Задача №061. Цифровая	80
Задача №062. Выплата монетами	81
Задача №063. Длина отрезка	81
Задача №064. Шахматная доска.....	82
Задача №065. Кубы.....	83
Задача №066. Количество различных чисел и n^2	84

Задача №067. Количество общих элементов	85
Задача №068. Соединение массива	86
Задача №069. Наименьшее целое положительное	87
Задача №070. Перестановка элементов массива.....	88
Задача №071. Прямоугольник максимальной площади	89
Задача №072. Фигура.....	91
Задача №073. 8 ферзей	93
Задача №074. 4 ладьи.....	95
Задача №075. Арабские и римские цифры	97
Задача №076. Пополнение сосуд	98
Задача №077. Карточки.....	99
Задача №078. Сумма длинных чисел.....	101
Задача №079. Персиковое дерево	103
Задача №080. Генератор комбинаций	104
Задача №081. Количество островов	105
Задача №082. Произведение длинных чисел.....	107
Задача №083. Сумма всех чисел	110
Задача №084. Последовательность	111
Задача №085. Большие числа	111
Задача №086. Allies.....	112
Задача №087. Нарисовать Символ.....	113
Задача №088. Маршруты	116
Задача №089. Треугольник	117
Задача №090. Прогрессия.....	118
Задача №091. Многоугольник	119
Задача №092. День рождения	119
Задача №093. N мер углов.....	121
Задача №094. Остатки	122
Задача №095. Снова многоугольник.....	123
Задача №096. Загрузка файлов	124
Задача №097. Уравнение.....	126
Задача №098. Арифметическая последовательность .	127
Задача №099. Правильная дробь	129
Задача №100. RLE.....	130

Задача №101. Сумма делителей	131
Задача №102. Обратная сумма.....	131
Задача №103. Футбольная лига	133
Задача №104. Коробка для яиц.....	135
Задача №105. Умножения строки	136
Задача №106. Диапазоны	137
Задача №107. Многочлен.....	139
Задача №108. Трамп и заборы	141
Задача №109. Дроби.....	143
Задача №110. Взаимно простые	144
Задача №111. Головоломка	145
Задача №112. Налог.....	148
Задача №113. Сумма цифр числа	149
Задача №114. Количество нечетных и четных	149
Задача №115. DVD-диски	150
Задача №116. Проверка уравнений	153
Задача №117. Остатки	154
Задача №118. Два треугольника	155
Задача №119. Квадратура круга	157
Задача №120. Рекурсивная сумма цифр числа.....	159
Задача №121. Остаток от деления числа.....	160
Задача №122. GPS	161
Задача №123. Стройся	162
Задача №124. Второй раз	164
Задача №125. Саморазделимое число.....	165
Задача №126. Создание пар	166
Задача №127. Красота	168
Задача №128. Последовательность 49.....	170
Задача №129. Числа Фибоначчи.....	171
Задача №130. Делители	172
ИСПОЛЬЗОВАННЫЕ ЛИТЕРАТУРА	174

ПРЕДИСЛОВИЕ

Этот решебник не может быть использован в качестве учебного пособия здесь отсутствует теория.

Авторы предполагают, что его читатель уже знает некоторые базисные понятия в программировании, умеет объявлять переменные и может самостоятельно скомпилировать некоторые программы.

На самом деле, во всем этом нет какой-то особой необходимости. В наше время в Интернете можно найти массу интересных теоретических материалов по программированию на языках Free Pascal, Pascal ABC .Net и т.д., и по мере надобности читатель, если ему что-то непонятно, может к ним обращаться.

Все, что можно найти в сборнике – это максимально подробные разборы задач. Они представлены в достаточно свободном стиле: сначала задача анализируется, затем составляется алгоритм, дается описание. В конце разбора каждой задачи приводится код всей программы.

Все решения задачи в данной пособии не являются уникальными решениями, т.е. в любой задаче может найтись другие пути решения.

Электронные варианты листинга программ всех задач, дистрибутивы компиляторов языка программирования PascalABC, PascalABC .Net и Free Pascal вы можете скачать из облачной хранилище по ссылке:

<https://cloud.mail.ru/public/CNPN/cRcPPjo6i>

Авторы будут признательны за любое указание на присутствующие в решебнике ошибки, недостатки в изложении материала и т. п., как и за любые другие комментарии.

Задача №001. А в степени N

Дано число **a** и натуральное число **n**. Вычислить **aⁿ** (**a** в степени **n**).

Решения:

Pascal ABC

Идея решения очевидна из Паскаль-кода.

```
program task_001;
var a,n,k: integer;
    b:longint;
begin
  readln(a,n);
  k:=0;b:=1;
  while k<>n do
  begin
    k:=k+1;
    b:=b*a;
  end;
  write(b);
end.
```

Задача №002. Число действий порядка $\log_2 n$

Решить задачу 001, если требуется, чтобы число действий (выполняемых операторов присваивания) было порядка **$\log_2 n$** (то есть не превосходило бы $C (\log_2 n)$, где C - константа, $\log_2 n$ - это степень, в которую нужно возвести 2, чтобы получить n).

Решения:

Pascal ABC

Идея решения очевидна из Паскаль-кода.

```

program task_002;
var a,n,k: integer;
    b,c:longint;
begin
readln(a,n);
k:=n; b:=1; c:=a;
while k<>0 do
begin
if k mod 2=0
then
begin
k:=k div 2;
c:=c*c;
end
else
begin
k:=k-1;
b:=b*c;
end;
end;
write(b);
end.

```

Задача №003. Вычисления факториалу

Дано натуральное число n , вычислить $n!$

Решения:

<i>Pascal ABC</i>

```

program task_003;
var n: integer;
    k,b:longint;
begin
readln(n);
k:=n;b:=1;

```

```

while k<>0 do
  begin
    b:=b*k;
    k:=k-1;
  end;
write(b);
end.

```

Задача №004. Вычисления суммы факториалов

Дано натуральное n , вычислить приближенно $1/1!+...+1/n!$ так, чтобы число операций (выполняемых команд присваивания) было бы порядка n (не более C_n для некоторой константы C).

Решения:

Pascal ABC

Важно не вычислять заново каждый раз $n!$ $n!=1\cdot2\cdot3\cdot...\cdot(n-1)\cdot n=(n-1)!\cdot n$. В нашем примере $1/n!=1/((n-1)!\cdot n)$, т.е. каждый j элемент получается делением $(j-1)$ -го элемента на j .

```

program task_004;
var n,k: integer;
    sum,last: real;
begin
  readln(n);
  k:=1;
  sum:=0;
  last:=1;
  while k<=n do
    begin
      sum:=sum+last;
      k:=k+1;
    end;
end.

```

```
    last:=last/k;  
end;  
writeln(sum);  
end.
```

Задача №005. Вычисление НОД

Даны два натуральных числа: **a** и **b**, не равные нулю одновременно. Вычислить **d=НОД(a,b)** – наибольший общий делитель **a** и **b**.

Решения:

<i>Pascal ABC</i>

Применяем Алгоритм Евклида: Будем считать, что

1) $\text{НОД}(0,0)=0$;

2) $\text{НОД}(a,b)=\text{НОД}(a-b,b)=\text{НОД}(a,b-a)$;

3) $\text{НОД}(a,0)=\text{НОД}(0,a)=a$ для всех $a, b \geq 0$.

```
program task_005;  
  var m,n,a,b,d:integer;  
begin  
  readln(a,b);  
  m:=a;n:=b;  
  while not ((m=0) or (n=0)) do  
    if m>=n then m:=m-n else n:=n-m;  
    if m=0 then d:=n else d:=m;  
  write(d);
```

Задача №006. НОД и уравнения

Даны два натуральных числа **a** и **b**, не равные нулю одновременно. Найти **d=НОД(a,b)** и такие **x** и **y**, что **d=ax+by**.

Решения:

Pascal ABC

Применяем Алгоритм Евклида как в задаче №006. Добавим переменные p, q, r, s такие, что $m=pa+qb, n=ra+sb$.

```
program task_006;
var a,b,n,m,p,q,r,s,k,x,y: integer;
begin
  read(a,b);
  m:=a; n:=b; p:=1;
  q:=0; r:=0; s:=1;
  while not ((m=0) or (n=0)) do
    if m>=n then begin m:=m-n; p:=p-r; q:=q-s; end
    else begin n:=n-m; r:=r-p; s:=s-q; end;
  if m=0 then begin k:=n; x:=r; y:=s; end
  else begin k:=m; x:=p; y:=q; end;
  writeln(x, ' ', y);
end.
```

Задача №007. Десятичные знаки числа

Даны натуральные числа n и $k, n>1$. Напечатать k десятичных знаков числа $1/n$. Программа должна использовать только целые переменные.

Решения:

Pascal ABC

Сдвинув в десятичной записи числа $1/n$ запятую на k мест вправо, получим число $(10^k)/n$. Нам надо напечатать его целую часть, то есть разделить 10^k на n нацело. Стандартный способ требует использования больших по величине чисел, которые могут выйти за границы диапазона представимых чисел.

Воспользуемся методом "деления уголком" и будем хранить "остаток" r .

```

program task_007;
var t,r,k,n: integer;
begin
  read(n,k);
  t:=0; r:=1;
  while t<>k do
  begin
    write((10*r) div n);
    r:=(10*r) mod n;
    t:=t+1;
  end;
end.

```

Задача №008. Функция

Функция f с натуральными аргументами и значениями определена так: $f(0)=0$, $f(1)=1$, $f(2n)=f(n)$, $f(2n+1)=f(n)+f(n+1)$. Составить программу вычисления $f(n)$ по заданному n , требующую порядка $\log_2 n$ операций (не более $C \cdot \log_2 n$).

Решения:

Pascal ABC

Функцию f можно записать в общем виде: $f(n)=a(f(k)+b(f(k+1)))$. Если n -четное, то $a=1$, $b=0$ иначе $a=1$, $b=1$.

Для $k=2m$: $f(k)=f(m)$, $f(k+1)=f(2m+1)=f(m)+f(m+1)$. Тогда $f(n)=af(k)+bf(k+1)=af(m)+b(f(m)+f(m+1))=(a+b)f(m)+bf(m+1)$. Для $k=2m+1$: $f(k)=f(m)+f(m+1)$, $f(k+1)=f(2m+2)=f(2(m+1))=f(m+1)$.

Тогда $f(n)=af(k)+bf(k+1)=af(m)+(a+b)f(m+1)$.

```

program task_008;
var a,b,k,n,m:integer;
begin

```

```

read(n);
k:=n; a:=1; b:=0;
while k<>0 do
  if k mod 2=0 then
    begin
      m:=k div 2;
      a:=a+b;
      k:=m;
    end
  else
    begin
      m:=k div 2;
      b:=a+b;
      k:=m;
    end;
write(b);
end.

```

{k=0: f(n)=af(0)+bf(1)=b, что и требовалось}

При каждом вхождении в цикл значение переменной k уменьшается вдвое. Поэтому число операций не более $3\log_2 n$.

Задача №009. Абракадабра

Последовательность из латинских букв строится следующим образом. На первом шаге она пуста. На каждом последующем шаге последовательность удваивается, после чего к ней слева дописывается очередная буква латинского алфавита (**a**, **b**, **c**, ...). Ниже приведены первые шаги построения последовательности:

Шаг 1. пустая последовательность

Шаг 2. a

Шаг 3. ba a

Шаг 4. cba a b a a

Шаг 5. dcba a b a a c b a a b a a

Задача состоит в том, чтобы по заданному числу N ($1 < N < 2^{26}$) определить символ, который стоит на N -ом месте в последовательности, получившейся после 26-го шага.

Решения:

Pascal ABC

Выполним обратные преобразования.

```
program task_009;
const len26=$1ffffff;
var
  n: longint;
  res: char;

procedure find(num:longint);
begin
  if n=1 then begin writeln(res); exit end;
  dec(num); dec(res); dec(n);
  num:=num shr 1;
  if n>num then n:=n-num;
  find(num);
end;

begin
  readln(n);
  if n>len26 then writeln('Введенное число больше ', len26)
  else
    begin
      res := 'y';
      find(len26);
    end;
end.
```

Задача №010. Количество различных чисел

Дан массив **x**: **array[1..n] of integer** ($1 < N < 1000$), причем **x[1] ≤ x[2] ≤ ... ≤ x[n]**. Найти количество различных чисел среди элементов массива.

Решения:

Pascal ABC

```
program task_010;
const z=1000;
var x:array[1..z] of integer;
    i,k,n:integer;
begin
  read(n);
  for i:=1 to n do read(x[i]);
  i:=1; k:=1;
  while i<>n do
  begin
    i:=i+1;
    if x[i]<>x[i-1] then k:=k+1;
  end;
  write(k);
end.
```

Задача №011. Протокол

Тройка чисел (**T1, M1, C1**) задает стартовое время, а тройка (**T2, M2, C2**) – финишное время участника лыжной гонки **30 км** (часы, минуты, секунды). Проверить корректность данных и найти результат участника.

Решения:

Pascal ABC

Идея решения очевидна из Паскаль-кода с комментариями.

```

program task_011;
var a,b,x,y,t1,m1,c1,t2,m2,c2 : integer;
begin
  write('старт...'); { ВВОД ДАННЫХ }
  readln(t1, m1, c1);
  write('финиш...');
  readln(t2, m2, c2);
  if (t1<0) or (t2<0) or (m1<0) or (m2<0) or {проверяем
корректность:}
(c1<0) or (c2<0) or (t1>23) or (t2>23) or {часы от 0 до
23,мин,сек- }
(m1>59) or (m2>59) or (c1>59) or (c2>59) { от 0 до 59 }
  then writeln('некорректные данные')
  else begin
    x:=(c2-c1) div 60;
    y:=(c2-c1) mod 60; {секунды}
    a:=((m2-m1) mod 60) + x; {МИНУТЫ}
    x:=(m2-m1) div 60;
    b:=((t2-t1) mod 60) + x; {часы}
    if y<0
      then begin
        y:=y+60;
        a:=a-1;
      end;
    if a<0
      then begin
        a:=a+60;
        b:=b-1;
      end;
    if b<0
      then b:=b+24;
    writeln('результат ',b,' ч. ',a,' мин. ', y,' сек. ');
  end;
end.

```

Задача №012. Длинная арифметика

Найти все цифры десятичной записи числа N ($0 \leq N \leq 10^{100}$).

Решения:

Pascal ABC

Задача на “длинную арифметику”. Моделируем умножение “в столбик”.

Умножаем $3*3*3*...*3$ (512 раз). Все данные представляем в виде строк, состоящих из цифр.

Функция Mult производит умножение числа, записанного в data, на однозначное число, записанное в C.

Для каждой цифры (справа налево) производится умножение на цифру C, результат записывается в переменную L. Остаток от деления $L/10$ записывается в результат, а частное прибавляется к следующему разряду.

```
program task_012;
var s1,s2,s,res,s0:string;
    a,k,n,i,j,m:integer;
function mult(data:string; c:char):string;
var i,k,l,m,f:integer;
    code : integer;
    s : string;
    d : string[1];
begin
    s:=data;
    val(c,m,code);
    f:=0;
    for i:=length(s) downto 1 do
        begin
            d:=s[i];
            val(d,k,code);
```

```

    l:=k*m+f;
    d:=inttostr(l mod 10);
    s[i]:=d[1];
    f:=l div 10;
end;
if f>0 then
begin
    d:=inttostr(f);
    s:=d+s;
end;
mult:=s;
end;

begin
writeln('ВВЕДИТЕ ОСНОВАНИЕ...');
readln(a);
writeln('ВВЕДИТЕ ПОКАЗАТЕЛЬ...');
readln(k);
res:="";
str(a,s1);
res:=s1;
n:=length(s1);
for m:=1 to k-1 do
for i:=n downto 1 do
begin
    s:=mult(res,s1[i]);
    res:=s;
end;
writeln(res);
end.

```

Задача №013. Счастливые билеты

Написать программу определения количества билетов с **6**-значными номерами, у которых сумма первых

3 десятичных цифр равна сумме 3 последних десятичных цифр.

Решения:

Pascal ABC

Организуем вложенный цикл с шестью параметрами. Каждый параметр определяет цифры 6-значного числа. Первый параметр начинается от 1 до 9, так как нет число начинающиеся с нуля. Остальные параметры начинается от 0 до 9. Проверим условие что суммы первые 3 цифры и последние 3 цифры равны и организуем счетчик.

```
program task_013;
uses crt;
var i,j,k,a,b,c:shortint;
    count: integer;
begin
    count:=0;
    for i:=1 to 9 do
        for j:=0 to 9 do
            for k:=0 to 9 do
                for a:=0 to 9 do
                    for b:=0 to 9 do
                        for c:=0 to 9 do
                            if i+j+k=a+b+c then count:=count+1;
    writeln(count);
end.
```

Задача №014. Поиск палиндромов

Символьная строка содержит последовательность слов, разделенных пробелами. Найти все палиндромы - слова, которые читаются слева направо так же, как и справа налево

Решения:

Pascal ABC

Для всех слов данного текста выполняем такие действия: Функция Palyndrom логического типа определяет, является ли данное слово палиндромом таким образом: первая буква сравнивается с последней, вторая - с предпоследней, и так либо до конца, либо до первого несовпадения.

```
program task_014;
uses crt;
var s,buf:string;
    i:byte;
function palyndrom(buf:string):boolean;
    var i,k:byte;
        ok:boolean;
begin
    ok:=true; i:=1;
    k:=length(buf);
    while ok and (i<k) do
        begin
            ok:=(buf[i]=buf[k]);
            inc(i);
            dec(k);
        end;
    palyndrom:=ok;
end;
```



```

begin
  clrscr;
  writeln('введите исходную строку...');
  readln(s);
  i:=1;
  while i<=length(s) do
    begin
      while s[i] = ' ' do inc(i);
      buf:="";
      while (s[i] <> ' ') and (i<=length(s)) do
        begin
          buf:=buf+s[i];
          inc(i);
        end;
      if palyndrom(buf) then writeln(buf, ' - палиндром.');
```

```

    end;
  end.
end.
```

Задача №015. Курьеры

Хлестакова приглашали управлять департаментом. В первый день ему прислали 1000 курьеров, а в каждый следующий - в 2 раза больше, чем в предыдущий. Хлестаков согласился, когда прислали 30000 курьеров. На какой день это произошло? (Учесь, что Хлестаков слаб в умножении и делении).

Решения:

Pascal ABC

Типичная итерационная задача. В цикле выполняем действие, пока количество курьеров не достигло 30000, каждый следующий день увеличиваем их количество в 2 раза (без умножения), считая при этом количество дней.

```

program task_015;
```

```

uses crt;
var s,n:longint;
    i:byte;
begin
  clrscr;
  i:=0;
  n:=1000;
  s:=0;
  while s<=30000 do
  begin
    s:=s+n;
    i:=i+1;
    n:=n+n;
  end;
  writeln('на ',i,' день!');
end.

```

Задача №016. Треугольник

Вычислите, в какой координатной четверти расположен треугольник, образованный прямой, заданной уравнением $y=ax+b$, и осями координат.

Решения:

Pascal ABC

Если в уравнении прямой угловой коэффициент положительный, то прямая лежит либо во второй, либо в четвертой четверти, при этом если свободный член положительный, то во второй, а если отрицательный – то в четвертой. Значит там же находится и весь треугольник, так как он образован этой прямой и осями координат. Аналогично рассуждаем и для отрицательного углового коэффициента: при положительном свободном члене – первая четверть, а при отрицательном – третья.

```

program task_016;
uses crt;
var a,b :real;
begin
  clrscr;
  write('y=a*x+b: введите a и b ...');
  readln(a,b);
  if (a=0) or (b=0)
    then writeln('треугольника не существует!')
    else begin
  if (a>0) and (b>0)
    then writeln('треугольник находится в II четверти. ');
  if (a>0) and (b<0)
    then writeln('треугольник находится в IV четверти. ');
  if (a<0) and (b>0)
    then writeln('треугольник находится в I четверти. ');
  if (a<0) and (b<0)
    then writeln('треугольник находится в III четверти. ');
    end;
end.

```

Задача №017. Попал-не-попал

Пусть D – заштрихованная часть плоскости (верхний полукруг радиуса 0.6 с центром в точке O , из которого вырезана правая верхняя четверть круга с тем же центром радиуса 0.3). Функция определяется следующим образом:

$U=x+y$, если (x, y) принадлежит D ; $U=x-y$ в противном случае. Даны числа x, y . Найти U .

Решения:

Pascal ABC

Принадлежность точки с координатами (x,y) заштрихованной части плоскости определяется конъюнкцией условий $y>0$ (принадлежность точки

верхней полуплоскости) и "непопадания" в вырезанную часть и за пределы круга.

```
program task_017;
uses crt;
var x,y,u:real;
begin
  clrscr;
  writeln('введите исходные данные... ');
  readln(x,y);
  if (y>=0) and (x*x+y*y<=0.36) and ((x<=0) or ((x>=0) and
    (x*x+y*y>=0.09))) then u:=x+y else u:=x-y;
  writeln(u);
end.
```

Задача №018. Произведение

Дано натуральное число **N**. Вычислить приближенно произведение первых **N** сомножителей.

$(2/1)*(2/3)*(4/3)*(4/5)*.....$

Решения:

Pascal ABC

В цикле прибавляем 2 поочередно к числителю (переменная X) или к знаменателю (переменная Y), накапливая произведение (переменная res).

```
program task_018;
uses crt;
var res:real;
    i,n,x,y, flag: integer;
begin
  clrscr;
  write('n=...');
  readln(n);
```

```

res:=1;
flag:=0;
x:=2;
y:=1;
for i:=1 to n do
  begin
    res:=res*x/y;
    if flag=0 then y:=y+2
      else x:=x+2;
    flag:=1-flag;
  end;
writeln(res);
end.

```

Задача №019. Поиск различных цифр

Дано натуральное **n**. Сколько различных цифр встречается в его десятичной записи?

Решения:

Pascal ABC

"Вырезаем" из числа цифры по одной. Объявляем вспомогательный хэш-массив из 10 элементов, каждый из которых увеличивается на 1 при получении из числа цифры, равной его индексу. После завершения анализа числа остается посчитать число ненулевых элементов хэш-массива.

```

program task_019;
uses crt;
var n :longint;
    a : array [0..9] of byte;
    i,k :byte;
begin
  write('введите число...');

```

```

readln(n);
for i:=0 to 9 do a[i]:=0;
while n<>0 do
begin
  i:=n mod 10;
  inc(a[i]);
  n:=n div 10;
end;
k:=0;
for i:=0 to 9 do
  if a[i]<>0 then inc(k);
writeln('в числе ',k,' цифр. ');
end.

```

Задача №020. Сумма 17-ти

Даны действительные числа $x[1], x[2], \dots, x[17]$. Найти сумму значений $|x[i] - x[j]|$ при условии, что $1 \leq i < j \leq 17$.

Решения:

Pascal ABC

Сумму накапливаем в переменной **s**. Перебираем все возможные пары чисел (i, j) . Если $i < j$, то вычисляем $|x[i] - x[j]|$ и полученное значение прибавляем к s.

```

program task_020;
uses crt;
var a: array[1..17] of real;
    i,j: byte;
    s: real;
begin
  for i:=1 to 17 do
    a[i]:=random(50);
  for i:=1 to 17 do
    write(a[i]:4);

```

```
writeln;
s:=0;
for i:=1 to 17 do
  for j:=1 to i-1 do
    s:=s+abs(a[i]-a[j]);
writeln(s)
end.
```

Задача №021. Уравнение

Написать программу для решения уравнения $xу+x+y=1000$ в целых числах.

Решения:

Pascal ABC

Это уравнение может быть решено без программирования. Прибавим по 1 к каждой из частей. Левая часть раскладывается на два множителя:

$$(x+1)*(y+1)=1001;$$

но $1001=7*11*13;$

отсюда возможны только случаи:

$$x+1=1; y+1=1001;$$

$$x+1=-1; y+1=-1001;$$

$$x+1=7; y+1=143;$$

$$x+1=-7; y+1=-143;$$

$$x+1=11; y+1=91;$$

$$x+1=-11; y+1=-91;$$

$$x+1=13; y+1=77;$$

$$x+1=-13; y+1=-77;$$

$$y+1=1; x+1=1001;$$

$$y+1=-1; x+1=-1001;$$

$$y+1=7; x+1=143;$$

$$y+1=-7; x+1=-143;$$

$$y+1=11; x+1=91;$$

$$y+1=-11; x+1=-91;$$

$$y+1=13; x+1=77;$$

$$y+1=-13; x+1=-77;$$

На всякий случай приведем переборное решение.


```

program task_021;
uses crt;
var x,y: longint;
begin
  for x:=-1002 to 1000 do
    for y:=-1002 to 1000 do
      if x+y+x*y=1000
        then writeln('x = ',x,'; y = ',y);
    end;
  end.

```

Задача №022. Среднее арифметическое

m и **n** – натуральные числа, $m \leq n$. Найти среднее арифметическое чисел **a[1],...**, **a[m-1]**, **a[m+1],...**, **a[n]** (всех, кроме **a[m]**).

Решения:

Pascal ABC

Идея решения очевидна из предлагаемого Паскаль-кода.

```

program task_022;
uses crt;
const max=20;
var a:array[1..max] of real;
    i,l:byte;
    s:real;
begin
  clrscr;
  randomize;
  for i:=1 to max do
    begin
      a[i]:=random*max;
      writeln(a[i]:15:7);
    end;
end;

```

```

write('l=...');
readln(l);
s:=0.0;
for i:=1 to max do
  if i<>l then s:=s+a[i];
s:=s/(max-1);
writeln('среднее арифметическое всех чисел, кроме ',l,'
равно ',s:15:7);
end.

```

Задача №023. Макси элемент

Дана целочисленная прямоугольная таблица **A[1:100,1:50]**. Найдите наибольшее из чисел, встречающихся в этой таблице.

Решения:

Pascal ABC

Идея решения очевидна из предлагаемого Паскаль-кода.

```

program task_023;
uses crt;
const m=10;
      n=5;
var a: array [1..m,1..n] of integer;
    i,j : byte;
    max : integer;
begin
  clrscr;
  randomize;
  for i:=1 to m do
    for j:=1 to n do
      a[i,j]:=random(30)-10;
  max:=a[1,1];

```

```

for i:=1 to m do
  for j:=1 to n do
    if a[i,j]>max then max:=a[i,j];
write('максимальный элемент таблицы равен ', max);
end.

```

Задача №024. Первый ноль

Дана целочисленная таблица **A[1:100]**. Проверьте, есть ли в ней элементы, равные нулю. Если есть, найдите номер первого из них, т.е. наименьшее **i**, при котором **A[i]=0**.

Решения:

Pascal ABC

Идея решения очевидна из предлагаемого Паскаль-кода.

```

program task_024;
uses crt;
const n=10;
var a: array [1..n] of integer;
    flag : boolean;
    i,k : integer;
begin
  clrscr;
  randomize;
  for i:=1 to n do
    a[i]:=random(10)-3;
  flag:=false; k:=0; i:=1;
  while (i<=n) and flag=false do
    begin
      if a[i]=0 then begin flag:=true; k:=i end;
      inc(i);
    end;
end;

```

```

if flag
  then
    write('первый из нулевых элементов таблицы имеет
номер ',k)
  else
    write('нулевых элементов в таблице нет ');
end.

```

Задача №025. Дальний минус

Дана целочисленная таблица $A[1:100]$. Проверьте, есть ли в ней отрицательные элементы. Если есть, найдите наибольшее i , при котором $A[i] < 0$.

Решения:

Pascal ABC

Задача тривиальна, однако красивое решение - просматривать массив с конца до встречи с отрицательным элементом и прекратить поиск.

```

program task_025;
uses crt;
const n=10;
var a: array [1..n] of integer;
    flag: boolean;
    i, k: integer;
begin
  clrscr;
  randomize;
  for i:=1 to n do
    a[i]:=random(10)-3;
  flag:=false; k:=0; i:=n;
  while (i>=1) and (flag=false) do
    begin
      if a[i]<0 then begin flag:=true; k:=i; end;

```

```

    dec(i);
end;
if flag
then
    write('последний из отрицательных элементов
таблицы имеет номер ', k)
else
    write('отрицательных элементов в таблице нет');
end.

```

Задача №026. Больше семи

n – натуральное, **a[1], a[2],..., a[n]** – целые. Заменить все большие 7 члены последовательности числом 7. Найти количество таких членов.

Решения:

Pascal ABC

Идея решения очевидна из приведенного Паскаль-кода.

```

program task_026;
uses crt;
const n=100;
var a: array[1..n] of integer;
    i, count: integer;
begin
    clrscr;
    count:=0;
    randomize;
    for i:=1 to n do
    begin
        a[i]:=random(17);
        write(a[i]:4);
    end;
end;

```

```
writeln;
for i:=1 to n do
  begin
    if a[i]>7 then
      begin
        a[i]:=7;
        inc(count);
      end;
    write(a[i]:4);
  end;
writeln('всего ', count);
end.
```

Задача №027. Делители

Найти все делители натурального числа **n**.

Решения:

Pascal ABC

Задача решается "в лоб". Делим число **n** последовательно на все числа от **1** до **n** и печатаем те, на которые число **n** делится без остатка.

```
program task_027;
var n, i:longint;
begin
  readln(n);
  i:=1;
  while i<=n do
    begin
      if n mod i = 0 then write(i, ' ');
      inc(i);
    end;
end.
```

Задача №028. Простые делители

Найти все простые делители натурального числа n .

Решения:

Pascal ABC

Напрашивается очевидное решение: найти все делители (см. задачу 17) и выбрать из них простые числа (как?). Однако красивее поступить иначе. Делим число n на 2, пока делится нацело, если разделилось хоть раз - число 2 - делитель (простой!), а n присваиваем результат деления. Увеличиваем делитель на 1 и начинаем сначала. Так поступаем до тех пор, пока число не станет равным 1.

```
program task_028;
uses crt;
var n, i: longint;
begin
  clrscr;
  write('введите число...');
  readln(n);
  write('делители числа ',n,' : ');
  i:=2;
  while n<>1 do
  begin
    if n mod i = 0 then
      begin
        write(i, ' ');
        repeat
          n:=n div i
        until n mod i <> 0;
        end;
      inc(i);
    end;
  end.
```


Задача №029. Замена

В таблице **a[1:100]** записаны нули и единицы. Заменить **0** на **1**, а **1** на **0**.

Решения:

Pascal ABC

Кажется, все очевидно: просматриваем массив. Если встретился 0, то заменяем его на 1, если встретилась 1 - заменяем на 0. Однако есть более оптимальное решение: не используя команды ветвления, заменить все элементы массива по формуле $a[i]:=1-a[i]$. Быстрее и красивее!

```
program task_029;
uses crt;
var a: array[1..100] of byte;
    i: byte;
begin
  clrscr;
  randomize;
  for i:=1 to 100 do
  begin
    a[i]:=random(2);
    write(a[i]);
  end;
  writeln;
  for i:=1 to 100 do
  begin
    a[i]:=1-a[i];
    write(a[i]);
  end;
end.
```

Задача №030. Таблица

Дана целочисленная таблица $A[1:1000]$. Подсчитайте наибольшее число идущих в ней подряд одинаковых элементов.

Решения:

Pascal ABC

Двигаясь по массиву, сравниваем рядом стоящие элементы. Если они равны-увеличиваем счетчик равных элементов, если нет - запоминаем, сколько было, и начинаем искать новую цепочку. Если новая цепочка окажется длиннее, запомним ее и т.д. до конца массива.

```
program task_020;
uses crt;
const n=10;
var a: array [1..n] of integer;
    i, k, kvo: integer;
begin
  clrscr;
  randomize;
  for i:=1 to n do
    a[i]:=random(3);
  kvo:=1;k:=1;
  for i:=2 to n do
    begin
      if a[i]<>a[i-1] then k:=1 else inc(k);
      if k>kvo then kvo:=k;
    end;
  write('в массиве есть ',kvo,' одинаковых подряд идущих
элементов');
end.
```

Задача №031. Сумма ряда №1

Даны натуральное n и вещественное a . Приблизительно вычислить $1+1/(a+1) + 1/(a+2)+ \dots + 1/(a+n)$.

Решения:

Pascal ABC

Задача тривиальна. Идея решения очевидна из приведенного Паскаль-кода.

```
program task_031;
var s, a, x: real;
    i, n: integer;
begin
  write('a=...');
  readln(a);
  write('n=...');
  readln(n);
  x:=a;
  s:=0;
  for i:=0 to n do
    begin
      x:=x*(a+i);
      s:=s+1/x;
    end;
  writeln('s=', s);
end.
```

Задача №032. Сумма ряда №2

Даны вещественные числа $a[1], \dots, a[n]$. Известно, что среди них есть отрицательные. Пусть первый среди отрицательных членов имеет номер $a[k+1]$. Вычислить $a[1]+2*a[2]+2*a[3]+ \dots +2*a[k-1]+a[k]$.

Решения:

Pascal ABC

Пока не встретилось отрицательное число, выполняем вычисления по формуле с учетом отсутствия удвоения у первого и последнего элемента суммы.

```
program task_032;
uses crt;
const n=100;
var a: array[1..n] of integer;
    i, k, s: integer;
begin
  clrscr;
  randomize;
  for i:=1 to n do
  begin
    a[i]:=random(40);
    a[i]:=a[i]-20;
    write(a[i]:4);
  end;
  writeln;
  i:=1;
  s:=0;
  while a[i]>=0 do
  begin
    if i=1 then s:=s+a[1]
      else s:=s+2*a[i];
    inc(i);
  end;
  if i>2 then s:=s-a[i-1];
  writeln(s);
end.
```

Задача №033. Выборы

На избирательном участке в списке из 100 избирателей указываются фамилия и название улицы, на которой проживает избиратель. Определить, сколько избирателей живет на улице Богуна.

Решения:

Pascal ABC

Основная идея – удачный выбор структуры данных. Создадим пользовательский тип DATA – запись с двумя строковыми полями и сосчитаем, в скольких из них после ввода данных поле Street равно “Богуна”.

```
program task_033;
const n=10;
type data = record
    name, street : string;
end;
var a: array[1..n] of data;
    count, i: byte;
begin
    count:=0;
    for i:=1 to n do
        begin
            writeln('Избиратель ',i);
            writeln('Имя...');
            readln(a[i].name);
            writeln('Улица...');
            readln(a[i].street);
            if a[i].street = 'Богуна' then
                begin
                    writeln(a[i].name,' живет на Богуна.');
```

```
end;  
writeln('Всего ', count, ' избирателей живут на Богуна.')  
end.
```

Задача №034. Деление на 13 или 17

Составить алгоритм, который заносит в таблицу первые **1000** натуральных чисел, делящихся на **13** или **17**.

Решения:

Pascal ABC

Пока число чисел, которые нацело делятся на 13 или 17, не достигло заданного, перебираем все натуральные числа, начиная с 13, и если число делится на 13 или 17, увеличиваем счетчик, а само число записываем в массив.

```
program task_034;  
uses crt;  
var count: integer;  
    x: longint;  
    a: array[1..1000] of longint;  
begin  
    x:=13;  
    count:=1;  
    while count<= 1000 do  
        begin  
            while (x mod 13<>0) and (x mod 17<>0) do inc(x);  
            writeln(x);  
            a[count]:=x;  
            inc(count);  
            inc(x);  
        end;  
    end.  
end.
```

Задача №035. Муравей

Посередине клетчатого листа бумаги нарисована замкнутая не самопересекающаяся ломаная, звенья которой идут по сторонам клеток. Муравей может переходить на одну из четырех соседних клеток, отмечать клетки, где он уже был. Муравей видит, пересек ли он линию и вышел ли он на край листа. Написать алгоритм, который определяет, где находится муравей: внутри области, ограниченной ломаной, или снаружи ее.

Технические условия:

Данные хранятся во внешнем файле **mur.dat** следующим образом: в первой строке записывается количество звеньев ломаной; вторая строка – координаты муравья; в каждой последующей строке записывается четыре числа – координаты двух клеток, между которыми проходит граница.

Образец текстового файла mur.dat

18	3 3 3 4	5 1 6 1
4 2	3 3 4 3	5 1 5 0
0 1 1 1	3 2 4 2	4 1 4 0
1 1 1 2	4 1 4 2	3 1 3 0
1 2 2 2	4 2 5 2	2 1 2 0
1 3 2 3	5 2 5 3	1 1 1 0.
2 3 2 4	5 2 6 2	

Решения:

Pascal ABC

Пусть муравей движется с исходной точки до края листа по прямой в любом направлении. Если количество пересечений линии было равно четному числу, то муравей находился вне области, ограниченной линией (0 – четное число!). Если же муравей пересечет линию нечетное число раз – он был в очерченной области. Решение базируется на том, что завершение пути муравья

всегда совпадает с краем листа, т.е. находится вне очерченной области (по условию задачи).

```
program task_035;
const maxsize=100;
type dat=array[1..maxsize,1..maxsize] of byte;
type coor=record
    x, y: real;
end;
var a:dat;
    i, j, ax, ay, bx, by: byte;
    f: text;
    p1,p2,q1,q2 :coor;
count, coll: integer;

function max(x,y:real):real;
begin
    if x>=y then max:=x
        else max:=y;
end;

function where(p,q,z:coor):integer;
{выдает -1, если точка z ниже прямой pq;
 0, если на прямой; 1, если выше}
var a,b,c,r: real; {это коэффициенты уравнения
ax+by+c=0}
begin
    a:=q.y-p.y;
    b:=p.x-q.x;
    c:=p.y*(q.x-p.x)-p.x*(q.y-p.y);
    r:=a*z.x+b*z.y+c;
    if r>0 then where:=1
        else if r<0 then where:=-1 else where:=0;
end;
```



```

function per(a1,b1,a2,b2:coor):byte;
{1 yes 0 no}
begin
  if (where(a1,b1,a2)*where(a1,b1,b2)=-1) and
    (where(a2,b2,a1)*where(a2,b2,b1)=-1)
    then per:=1
    else per:=0;
end;

```

```

begin
  count:=0;
  assign(f,'mur.dat');
  reset(f);
  readln(f,coll);
  readln(f, q1.x, q1.y);
  q1.x:=q1.x+0.5;
  q1.y:=q1.y+0.5;
  q2.x:=0.5;
  q2.y:=0.0;
  for i:=1 to coll do
    begin
      readln(f,ax,ay,bx,by);
      if ax=bx then
        begin {горизонтальная линия}
          if abs(ay-by)=1 then
            begin
              p1.y:=max(ay,by);
              p2.y:=p1.y;
              p1.x:=ax;
              p2.x:=ax+1;
            end
          end
        end
      else if ay=by then
        begin {вертикальная линия}
          if abs(ax-bx)=1 then

```

```

begin
  p1.x:=max(ax,bx);
  p2.x:=p1.x;
  p1.y:=ay;
  p2.y:=ay+1;
end
end; {иначе неверная строка в файле}
count:=count+per(p1,p2,q1,q2);
end;
close(f);
if odd(count) then writeln('Внутри!')
  else writeln('Снаружи!')
end.

```

Задача №036. Подсчет чисел

Дано 100 чисел от 1 до 50. Определить, сколько среди них чисел Фибоначчи и сколько чисел, первая значащая цифра в десятичной записи которых 1 или 2.

Решения:

<i>Pascal ABC</i>

Идея решения очевидна из исходного текста с комментариями.

```

program task_036;
var i,s,x: integer;
    prost: set of 1..50;
    a: array [1..100] of integer;

{Находим числа Фибоначчи}
function fib(n:integer): integer;
begin
  if (n=1)or(n=2) then fib:=1 else fib:=fib(n-2)+fib(n-1);
end; {fib}

```

{Заполняем множество числами Фибоначчи<50}

```
procedure setinit;
```

```
  var flag: boolean;
```

```
      f, m: integer;
```

```
begin
```

```
  prost:=[];m:=1;flag:=true;
```

```
  while flag do
```

```
    begin
```

```
      f:=fib(m);
```

```
      if f>50 then flag:=false else prost:=prost+[f];
```

```
      m:=m+1
```

```
    end;
```

```
end; {setinit}
```

```
begin
```

```
  setinit; {Заполнение множества числами Фибоначчи}
```

```
  randomize; {Заполняем массив числами<=50}
```

```
  for i:=1 to 100 do a[i]:=random(50);
```

```
  s:=0; x:=0;
```

```
  for i:=1 to 100 do
```

```
    begin
```

```
      if a[i] in prost then s:=s+1; {если элемент массива-число  
Фибоначчи}
```

```
      if (a[i] div 10=1) or (a[i] div 10=2)
```

```
then x:=x+1 {если первая цифра 1 или 2, а число двухзначное}
```

```
else if a[i] div 10=0
```

```
then if (a[i] mod 10=1)or(a[i] mod 10=2) {если однозначное}
```

```
      then x:=x+1;
```

```
    end;
```

```
  writeln('В массиве ',s,' чисел Фибоначчи');
```

```
  writeln('В массиве ',x,' элементов с 1 или 2 в начале');
```

```
end.
```

Задача №037. Три числа

Написать программу, которая по заданным трем числам определяет, является ли сумма каких-нибудь двух из них положительной.

Решения:

Pascal ABC

Задача тривиальна. Алгоритм очевиден из приведенного Паскаль-кода программы.

```
program task_037;
var a,b,c: real;
begin
  writeln('Введите числа...');
  readln(a,b,c);
  if (a+b>0) or (a+c>0) or (b+c>0)
  then writeln('yes')
  else writeln('no')
end.
```

Задача №038. Поиск чисел №1

Написать программу, которая находит и выводит на печать все четырехзначные числа **abcd**, для которых выполняются следующие условия:

1) **a, b, c, d** – разные цифры; 2) $ab - cd = a + b + c + d$.

Решения:

Pascal ABC

Организуем 4 вложенных цикла по количеству разрядов числа. Первый цикл - от 1 до 9, а три внутренних - от 0 до 9. А дальше... перебор, пока не выполнится условие.

```

program task_038;
var a, b, c, d: byte;
begin
  for a:=1 to 9 do
    for b:=0 to 9 do
      if a<>b then
        for c:=0 to 9 do
          if (a<>c) and (b<>c) then
            for d:=0 to 9 do
              if (a<>d) and (b<>d) and (c<>d)
                then if a*b-c*d=a+b+c+d then write(a,b,c,d, ' ');
end.

```

Задача №039. Поиск чисел №2

Написать программу, которая находит и выводит на печать все четырехзначные числа $abcd$, для которых выполняются следующие условия:

1) a, b, c, d - разные цифры

2) $\overline{ab} - \overline{cd} = a + b + c + d$

(\overline{ab} означает, что число состоит из цифр a и b)

Решения:

Pascal ABC

Задача аналогична предыдущей. Отличие заключается в проверке условия.

```

program task_039;
uses crt;
var a,b,c,d: byte;
begin
  clrscr;
  for a:=1 to 9 do

```

```

for b:=0 to 9 do
  if a<>b then
    for c:=0 to 9 do
      if (a<>c) and (b<>c) then
        for d:=0 to 9 do
          if (a<>d) and (b<>d) and (c<>d)
            then if (10*a+b)-(10*c+d)=a+b+c+d
              then write(a,b,c,d, ' ');
end.

```

Задача №040. Самая длинная цепочка

Дана вещественная таблица $a[1], a[2], \dots, a[1000]$. Определить максимальное количество подряд идущих положительных элементов последовательности, не прерываемых ни нулями, ни отрицательными элементами. Напечатать найденный фрагмент.

Решения:

Pascal ABC

Движемся вдоль массива. Фиксируем начало и конец “положительного” фрагмента и запоминаем его длину, индексы первого и последнего элементов. Аналогично поступаем со следующим фрагментом (если он есть!). Если следующий фрагмент длиннее предыдущего, то помним его начало, конец и длину. Так (за один проход!) проходим по всему массиву и на финише имеем то, что позволит нам напечатать интересующий нас фрагмент. Комментарии в исходном тексте позволят до конца уяснить решение.

```

program task_040;
uses crt;
const max=100; {сколько элементов в массиве}
var a: array[1..max] of integer; {исследуемый массив}

```

```

    maxstart, maxstop, i, len, start, stop: integer;
begin
  clrscr;
  randomize; {заполняем массив случайными числами}
  for i:=1 to max do
    begin
      a[i]:=random(5);
      a[i]:=a[i]-1; {чтобы были числа меньше 0}
      write(a[i], ' ');
    end;
  i:=1;
  maxstart:=0; {начало искомого фрагмента}
  maxstop:=0; {конец искомого фрагмента}
  len:=0; {текущая длина фрагмента}
  while i<=max do
    begin
      while (i<=max) and (a[i]<=0) do {ищем начало
фрагмента...}
        inc(i);
      start:=i; {запоминаем начало фрагмента}
      while (i<=max) and (a[i]>0) do {ищем конец фрагмента}
        inc(i);
      stop:=i-1; {запоминаем конец фрагмента}
      if len<stop-start
      then
        begin {текущий фрагмент длиннее всех предыдущих}
          len:=stop-start;
          maxstart:=start;
          maxstop:=stop;
        end;
    end;
  writeln;
  if maxstop=0
  then writeln('Нет положительных элементов')
  else begin

```

```
writeln('Найдена последовательность от ', maxstart,
' до ', maxstop);
for i:=maxstart to maxstop do
begin
write(a[i], ' ');
end;
end;
end.
```

Задача №041. Сообщество роботов

Сообщество роботов живет по следующим законам: – один раз в начале года они объединяются в группы по три или пять роботов;

- за год группа из **3** роботов собирает **5** новых, а группа из **5** роботов собирает **9** новых;
- роботы собираются так, чтобы собрать за год наибольшее количество;
- каждый робот живет три года после сборки.

Известно начальное количество роботов **k** и все они только что собраны. Сколько роботов будет через **n** лет?

Решения:

Pascal ABC

5 группы по 3 робота = 15 – собирают **25**

3 группы по 5 роботов = 15 – собирают **27**

27>25

В оптимальном разбиении роботов на группы нет 5 групп по 3 робота.

- **a1**- количество собранных роботов на 1 й год.
- **a2=f(a1)** количество собранных роботов на 2 й год.
- **a3=f(a1+a2)** количество собранных роботов на 3 й год.

- $a_4=f(a_1+a_2+a_3)$ количество собранных роботов на 4 й год.
- $a_5=f(a_2+a_3+a_4)$ количество собранных роботов на 5 й год.

Здесь a_1 не участвует потому что роботы живет только 3 года.

- **pp** – количество собранных роботов перед перед данного года
- **pr** – количество собранных роботов перед данного года
- **cur** – количество собранных роботов в данный год;
- **nt**- количество собранных роботов после данного года.

$nt=f(pp+pr+cur)$

$pp<-pr$

$pr<-cur$

$cur<-nt$

в начало будет: $pp=0; pr=0; cur=k$

```

program task_041;
uses Math;
var i,n:integer;
    pp,pr,cur,nt:int64;

function f(n:int64):int64;
var m,n3,n5:int64;
begin
    m:=0;
    n3:=0;
    while (n3<=4) and (n3*3<=n) do
    begin
        n5:=(n-n3*3) div 5;
        m:=max(m,n3*5+n5*9);
    end
end

```

```

        inc(n3);
    end;
    f:=m;
end;

begin
    pp:=0;
    pr:=0;
    readln(cur,n);
    for i:=2 to n do
        begin
            nt:=f(cur+pr+pp);
            pp:=pr;
            pr:=cur;
            cur:=nt;
        end;
        writeln(pp+pr+cur);
    end.

```

Задача №042. Физическая лаборатория

При проведении физического эксперимента по фиксации траектории движения частиц с помощью ЭВМ детекторы сгруппированы следующим образом:

```

Д Д ... Д
Д Д ... Д
.....
Д Д ... Д

```

Перекрытая детекторами область отображается в памяти ЭВМ двумерным массивом $\mathbf{M}[1:N,1:K]$, элементы которого представляют собой цифровую фотографию исследуемой области. При фиксации элементарной частицы детекторами в позиции (\mathbf{I}, \mathbf{J}) соответствующий элемент матрицы \mathbf{M} принимает значение $\mathbf{1}$, в противном случае $\mathbf{0}$. Определите, содержит ли данная цифровая

фотография информацию, которая может быть интерпретирована как след прямолинейной траектории, начинающейся и заканчивающейся за пределами фотографии.

Решения:

Pascal ABC

Выбираем показания 2-х сработавших детекторов и проверяем, не лежат ли другие сработавшие детекторы на одной прямой с ними. Если да, то это след прямолинейно двигавшейся частицы. Пусть координаты первого детектора (x_1, y_1) , а второго (x_2, y_2) . Найдем $dx = x_2 - x_1$ и $dy = y_2 - y_1$. Разделим dx и dy на их наибольший общий делитель. Для каждой точки с координатами (x, y) находим расстояние $(x - x_1)$ и $(y - y_1)$. Для того, чтобы эта точка находилась на прямой, проходящей через первый и второй детекторы, необходимо и достаточно, чтобы $(x - x_1)$ и $(y - y_1)$ были пропорциональны dx и dy . В массиве записей мы храним координаты сработавших детекторов.

```
program task_042;
const max = 100; {всего сработавших детекторов}
type coor = record {координаты сработавшего детектора}
    x, y: byte
end;

var b: array [1..max] of coor;
    n, i, dx, dy, z: integer;
    ok: boolean;

{функция возвращает НОД для x, y}
function nod(x,y: byte): byte;
var k: byte;
begin
    if x<y then begin k:=x; x:=y; y:=k; end;
```

```

if (x=0) or (y=0)
  then nod:=1
  else
    begin
      while x mod y <> 0 do
        begin
          k:=x mod y;
          x:=y; y:=k;
        end;
      nod:=y;
    end;
end;

begin
  write('Количество точек...');
  readln(n);
  for i:=1 to n do
    begin
      write('Координаты точки ', i, ': ');
      readln(b[i].x, b[i].y)
    end;
  dx:=b[1].x-b[2].x;
  dy:=b[1].y-b[2].y;
  z:=nod(abs(dx),abs(dy));
  dx:=dx div z;
  dy:=dy div z;
  i:=3; ok:=true;
  while ok and (i<=n) do
    begin
      ok:=(b[i].x-b[1].x)*dy = (b[i].y-b[1].y)*dx;
      inc(i);
    end;
  if ok then writeln('Траектория прямолинейна.')
  else writeln('Траектория не прямолинейна.')
end.

```

Задача №043. Счастливые номера

На интервале (1000;9999) найти все простые числа, каждое из которых обладает тем свойством, что сумма первой и второй цифры записи этого числа равна сумме третьей и четвертой цифр.

Решения:

Pascal ABC

В цикле проверяем число вначале на “простоту”, а затем на равенство сумм первых и последних двух цифр.

```
program task_043;
  var m, n, i: integer;

  {Находит сумму цифр числа x}
  function summ(x: integer): integer;
    var y, s, k: integer;
  begin
    s:=0; y:=x;
    while y<>0 do
      begin
        k:=y mod 10;
        y:=y div 10;
        s:=s+k
      end;
    summ:=s
  end; {summ}

  {Если x - простое, то “истина”...}
  function prost(x:integer): boolean;
    var ok: boolean;
        i: integer;
  begin
```

```

if x=2
  then ok:=true
  else if not odd(x)
    then ok:=false
    else
      begin
        ok:=true;
        i:=3;
        while ok and (i*i<=x) do
          begin
            ok:=(x mod i <> 0);
            i:=i+2;
          end;
        end;
      prost:=ok;
    end;

begin
  for i:=1000 to 9999 do
    if prost(i) then
      begin
        m:=i div 100;
        n:=i mod 100;
        if summ(m)=summ(n) then writeln(i)
      end
    end;
end.

```

Задача №044. Перевозка грузов

Из одного порта в другой необходимо перевезти **15** различных грузов. Грузоподъемность судна, на котором будет проходить перевозка, **50** тонн. Грузы пронумерованы, и информация о массах грузов хранится в массиве **M[15]**. Определить, сколько рейсов необходимо сделать судну, если грузы неделимы и могут перевозиться

только подряд в порядке их нумерации. (Предполагается, что масса отдельного груза не превышает **50** тонн).

Решения:

Pascal ABC

Обозначим: k - номер рейса судна, i - номер очередного груза, s - масса груза на судне в k -том рейсе. Решать задачу будем так: если на судно в k -том рейсе можно поместить ещё один груз, то мы грузим его и берём следующий, если груз не может быть размещен, то перевозим его следующим рейсом (увеличиваем k).

```
program task_044;
var m: array[1..15] of integer;
    k, s, i: integer;
begin
  k:=1; i:=1; s:=0;
  repeat if s+m[i]<=50
  then
    begin
      s:=s+m[i];
      i:=i+1;
    end
  else begin k:=k+1; s:=0; end;
until i>15;
writeln('Всего потребовалось ', k, ' рейсов');
end.
```

Задача №045. Знак #

Задана строка символов. Среди литер этого текста особую роль играет знак **#**, появление которого означает отмену предыдущей литеры текста; **k** знаков **#** отменяют **k** предыдущих литер (если такие есть) Напечатать строку

с учетом роли знака **#**. Например, строка "VR#Y##HELO#LO" должна быть напечатана в виде: "HELLO".

Решения:

Pascal ABC

Обозначим *s* - исходную строку, *len* - длину этой строки. Для решения создадим ещё одну строку *s1* (вначале пустую). Далее организуем цикл по номеру символа в строке *s*. Если очередной символ не **#**, то добавим его к строке *s1*, если это знак **#** и строка *s1* не пустая, то удалим из неё последний символ.

```
program task_045;
var s,s1: string;
    i, k, len: integer;
begin
  readln(s);
  len:=length(s);
  s1:="";
  k:=0;
  for i:=1 to len do
    if (s[i]='#')and(k<>0)
    then
      begin
        delete(s1,k,1);
        k:=k-1;
      end
    else
      begin
        k:=k+1;
        s1:=s1+s[i];
      end;
  writeln(s1);
end.
```


Задача №046. Произведение в строке

Для заданной строки символов вычислить произведение входящих в эту строку целых чисел (без учета их знаков). Например, для строки "kjjkkj2.5jkjn,,,hfd45jgfvjlkfdii10,2hfhg" произведение $2*5*45*10*2=9000$.

Решения:

Pascal ABC

Пусть s - строка. Обозначим длину строки - len . Организуем цикл, в котором будем проверять, является ли очередной символ цифрой, если да, то организуем новый цикл, в котором будем формировать строку sn , состоящую из цифр (очередное целое число). Потом преобразуем sn в число и вычислим произведение. Программа на Паскале, реализующая данный алгоритм, будет иметь следующий вид (переменная p в этой программе используется для накапливания значения произведения, переменная kod - для хранения кода результата преобразования строки в число):

```
program task_046;
var sn, s: string;
    len, k, kod: integer;
    v, p: real;
begin
  readln(s);
  len:=length(s);
  p:=1; k:=1;
  repeat
    sn:="";
    while (k<=len) and (s[k]>='0') and (s[k]<='9') do
      begin
```

```

    sn:=sn+s[k];
    k:=k+1;
end;
if sn<>" then
begin
    val(sn,v,kod);
    p:=p*v;
end;
k:=k+1;
until k>len;
writeln(' p=',p);
end.

```

Задача №047. Замена слова местами

Задана строка символов, содержащая два или более слов, разделенных пробелами. Написать программу, меняющую местами все четные и нечетные слова в строке, предполагая, что за один раз можно менять местами не более двух символов.

Решения:

Pascal ABC

Имеется несколько путей решения этой задачи. Для упрощения предположим, что строка не начинается и не заканчивается пробелом и что между словами в строке стоит ровно по одному пробелу. Пусть известна пара слов, которую необходимо переставить, и - номера первой и последней букв в первом слове, и - номера первой и последней букв во втором слове. Рассмотрим способ, в котором для перестановки слов будем использовать следующий алгоритм:

Запишем буквы первого слова в обратном порядке (поменяв первую с последней, вторую с предпоследней и

т.д.). Например, из строки **abcd efghi** получим **dcba efghi**. Потом аналогичным образом переставим буквы второго слова: из строки **dcba efghi** получим **dcba ihgfe**. Для получения окончательного результата необходимо записать буквы полученного словосочетания в обратном порядке: Из строки **dcba ihgfe** получим **efghi abcd** (что и требовалось получить).

Таким образом, для перестановки двух слов достаточно написать подпрограмму, которая меняет в заданной строке порядок букв на противоположный (инвертирует строку), и вызвать эту подпрограмму для первого слова, второго слова и всего словосочетания.

Обозначим `invert(k,l)` - процедуру, которая записывает в заданной строке `s` символы с `k`-того по `l`-й в обратном порядке, тогда программа, решающая задачу, будет иметь вид:

```

program task_047;
var s: string;
    i,n,m1,m2,l1,l2: byte;

procedure invert(k,l:byte);
var i:byte;
    ch:char;
begin
  for i:=k to ((l+k) div 2) do
  begin
    ch:=s[i];
    s[i]:=s[l+k-i];
    s[l+k-i]:=ch;
  end;
end;

```

```

begin
readln(s);
i:=0; n:=0;
m1:=1;m2:=1;l1:=1;l2:=1;
while i<length(s) do
begin
i:=i+1;
if (s[i]=' ')or(i=length(s))
then
begin
n:=n+1;
if n=1 then
begin
m2:=i-1;
l1:=i+1
end
else
begin
n:=0;
if i=length(s) then l2:=i else l2:=i-1;
invert(m1,m2);invert(l1,l2);invert(m1,l2);
m1:=i+1
end
end
end;
writeln(s)
end.

```

Задача №048. Площадь многоугольника

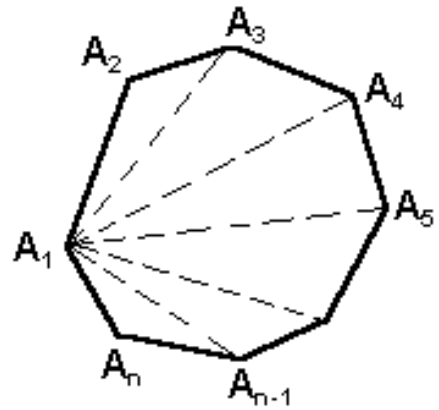
Выпуклый многоугольник задан последовательностью координат своих вершин в порядке обхода: (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , ..., (x_n, y_n) . Вычислить площадь многоугольника.

Решения:

Pascal ABC

Стандартный способ вычисления площади выпуклого многоугольника - разбиение исходного многоугольника на отдельные треугольники (рис.) с последующим вычислением площадей полученных треугольников и их суммированием.

Площадь отдельного треугольника можно вычислить, например, по формуле Герона, но в данном случае более удобной будет формула расчета площади треугольника по координатам его вершин:



$$S = \frac{1}{2} |(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)|$$

Пусть n - число вершин, $X(n)$, $Y(n)$ - массивы, содержащие координаты вершин, тогда основная часть программы для вычисления площади многоугольника будет иметь вид:

```
program task_048;
  const n=6;
  var i: integer;
      s: real;
      x, y: array[1..n] of integer;
begin
  s:=0;
  for i:=3 to n do
    readln(x[i], y[i]);
  for i:=3 to n do
    s:=s+0.5*abs((x[i-1]-x[1])*(y[i]-y[1])-(x[i]-x[1])*(y[i-1]-y[1]));
  writeln('Площадь многоугольника s=',s);
end.
```

Задача №049. Окружности

Даны две окружности, требуется сказать – пересекаются они или нет.

В первой строке задаются координаты центра первой окружности и радиус: **X1 Y1 R1**. Во второй строке – координаты центра и радиус второй окружности: **X2 Y2 R2** ($-1000 \leq X1, Y1, X2, Y2 \leq 1000, 1 \leq R1, R2 \leq 1000$).

Вывести “**YES**”, если пересекаются, “**NO**” не пересекаются.

Решения:

<i>Pascal ABC</i>

Применяя формулу вычисления расстояния между двумя точками $AB = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ найдем расстояния между центрами окружности. Сравниваем его суммами радиусов.

```
program task_049;
var x1, y1, r1, x2, y2, r2: shortint;
begin
  readln(x1, y1, r1);
  readln(x2, y2, r2);
  if sqrt(sqr(x2- x1)+ (y2- y1))<(r1+r2)
    then writeln('Yes')
    else writeln('No');
end.
```

Задача №050. Делители

Дано натуральное число **M**. Получить в порядке возрастания **N** первых натуральных чисел не больше, чем **M**, которые не делятся ни на какие простые числа кроме **2, 3, 5**. У числа должны быть все простые делители **2, 3 и 5**.

Например, число 30 является искомым числом, число имеет все простые делители $2*3*5=30$, а число 10 только 2 и 5 и потому не является искомым числом.

Первая строка содержит два целых числа M и N .
 $1 \leq M \leq 105$, $1 \leq N \leq 1000$.

Вывести N чисел, по одному в строке в порядке возрастания. Если таких чисел меньше, чем N , то вывести их все. Если таких чисел нет, то вывести 0.

Решения:

Pascal ABC

Алгоритм очевиден из приведенного Паскаль-кода программы.

```
program task_050;
var m, n, i, k: shortint;
begin
  readln(m,n);
  for i:=1 to m do
    if (i mod 2=0) and (i mod 3=0) and (i mod 5=0) and (k<n)
      then
        begin
          writeln(i);
          inc(k);
        end;
    if k<n then writeln('0');
  end.
```

Задача №051. Дробь

Дана дробь, записанная в виде числителя и знаменателя. Требуется преобразовать ее в правильную дробь и сделать несократимой. Два числа даются через пробел, M и N – числитель и знаменатель дроби.
 $1 \leq M, N \leq 10^9$.

Вывести целую часть дроби, а после, через пробел числитель и знаменатель, разделенные знаком “/”. Если число не имеет дробной части, выводить дробную часть не надо. Если число не имеет целой части, целую часть и пробел выводить не надо.

Например, $10/3 \rightarrow 3 \ 1/3$

$100/60 \rightarrow 1 \ 2/3$

Решения:

Pascal ABC

Алгоритм заключается в следующем. Сначала найдем наибольший общий делитель заданных чисел т.е. НОД(M,N). Дробью $\frac{M:\text{НОД}(M,N)}{N:\text{НОД}(M,N)}$ является несократимым дробям. Из этой дроби найдем целую часть и выведем числителя и знаменателя.

```
program task_051;
var n,m,x,y:int64;

function gcd(a,b: int64): int64;
var t: int64;
begin
  while b <> 0 do
  begin
    t:=b;
    b:=a mod b;
    a:=t
  end;
  gcd:=a
end;

begin
  readln(m,n);
  if m>=n then
```



```

write(m div n);
if (m>=n) and (m mod n<>0) then
  write(' ');
m:=m mod n;
x:=m div gcd(m,n);
y:=n div gcd(m,n);
if m<>0 then write(x,'/',y);
end.

```

Задача №052. Большие числа

Вывести на экран цифры числа 3^{1000} . Если попытаться получить число непосредственно умножением, компьютер выдаст сообщение об ошибке.

Решения:

Pascal ABC

Для записи больших чисел удобно использовать массивы, записывая цифры числа как элементы массива. Оценим количество цифр, необходимых для записи 3^{1000} :

$$3^2 = 9; 3^{1000} = (3^2)^{500} = 9^{500} \approx 10^{500}.$$

Таким образом, в записи этого числа будет менее 500 цифр. Запишем вначале в массив только число 3 и далее будем умножать его поэлементно на 3 нужное число раз, учитывая перенос из разряда в разряд, возникающий при умножении.

```

program task_052;
const stp=1000;
var i,j,k,prn,x:integer;
    a:array [1..500] of integer;
begin
  a[500]:=3; prn:=0;
  for i:=2 to stp do

```

```

for j:=500 downto 1 do
  begin x:=a[j]*3;
        a[j]:=(x+prn) mod 10;
        prn:=(x+prn) div 10;
  end;
k:=1; while (a[k]=0) do k:=k+1;
for i:=k to 500 do write(a[i]:1);
end.

```

Задача №053. Игра

Два третьеклассника играют в игру. Они берут слово и вычеркивают по очереди одной **четной** букве. Победителем считается тот, кто вычеркнет последнюю **четную** букву из данного слова. Ваша задача определить победителя. Первым ходит игрок номер 1, вторым игрок номер 2.

В строке дано слово состоящее из букв латинского алфавита длиной от 1 до 1000 символов. Если побеждает игрок 1, вывести **Player 1**, иначе **Player 2**.

Решения:

<i>Pascal ABC</i>

```

program task_053;
var s:string;
    len:integer;
begin
  read(s);
  len:=length(s);
  if (len mod 2=0) then write('Player 1')
  else write('Player 2');
end.

```

Задача №054. Черно-белый рисунок

Яркости пикселей рисунка закодированы числами от 0 до 255 в виде матрицы. Преобразовать рисунок в черно-белый по следующему алгоритму:

1) вычислить среднюю яркость пикселей по всему рисунку;

2) все пиксели, яркость которых меньше средней, сделать черными (записать код 0), а остальные – белыми (код 255).

Первая строка, содержит число N ($1 \leq N \leq 10$) – количество строк и столбцов в матрице. В следующих N строках содержатся по N элементов матрицы.

Вывести получившуюся матрицу.

Решения:

<i>Pascal ABC</i>

```
program task_054;
const k=10;
var n,i,j,s:integer;
    a:array[1..k,1..k] of integer;
    f,g:text;
begin
assign(f,'input.txt');
assign(g,'output.txt');
reset(f);
rewrite(g);
read(f,n);
for i:=1 to n do
for j:=1 to n do
begin
read(f,a[i,j]);
s:=s+a[i,j];
end;
s:=s div sqr(n);
```

```

for i:=1 to n do
for j:=1 to n do
if a[i,j]<s then a[i,j]:=0
else a[i,j]:=255;

```

```

for i:=1 to n do
begin
for j:=1 to n do
write(g,a[i,j],' ');
writeln(g,"");
end;
close(f);
close(g);
end.

```

Задача №055. Сумма больших чисел

Ввести два натуральных числа **A** и **B** ($1 \leq A, B \leq 10^{108}$) и вывести их сумму.

Решения:

Pascal ABC

```

program task_055;
uses crt;
const n=20;
var st1,st2: string[n];
    a,b,c:array[1..n] of integer;
    i,k,memo:integer;
    bool:boolean;
begin
readln(st1,st2);
k:=length(st1);
for i:=n downto n+1-k do a[i]:=strtoint(st1[(i-(n-k))]);
k:=length(st2);
for i:=n downto n+1-k do b[i]:=strtoint(st2[(i-(n-k))]);

```

```

memo:=0;
for i:=n downto 1 do
begin
  k:= a[i]+b[i]+memo;
  if k>=10 then begin c[i]:=k mod 10; memo:=k div 10; end
  else begin c[i]:=k; memo:=0; end;
end;
for i:=1 to n do
  if c[i]<>0 then begin k:=i; break; end;
for i:=k to n do write(c[i]);
end.

```

Задача №056. Алфавит

Удалить из данного “слова”, содержащего пять заглавных латинских букв, повторяющиеся буквы, и расположить оставшиеся буквы в алфавитном порядке.

Решения:

<i>Pascal ABC</i>

```

program task_056;
var buk: set of 'A'..'Z';
    s: string;
    i: integer;
    c: char;
begin
  readln(s);
  for i:=1 to length(s) do
  begin
    s[i]:=upcase(s[i]);
    include(buk, s[i]);
  end;
  for c := 'A' to 'Z' do
    if c in buk then write(c);
end.

```

Задача №057. Пары

Даны натуральное число N и N пар натуральных чисел. Найти в каждой паре минимальное число из всех таких чисел и найти максимальное число их.

Решения:

<i>Pascal ABC</i>

```
program task_057;
var n,k,a,b,c,max:integer;
begin
  k:=1;
  read(n);
  while k<=n do
  begin
    readln(a,b);
    if a<b then c:=a else c:=b;
    if c>max then max:=c;
    k:=k+1;
  end;
  write(max);
end.
```

Задача №058. Номер страницы

Однажды робот-библиотекарь решил устроить ревизию. На одной из полок, среди экземпляров тридцать третьего издания Кормена, он нашел листок из условий одного древнего контеста. Роботу известен формат оформления условий, однако этот листок привел его в замешательство.

Обычно внизу каждой страницы условий есть надпись вида «Страница i из « n », где i — номер страницы условий, а n — количество страниц в условиях. Однако на

этом листе была всего одна длинная последовательность цифр. Видимо, принтер почему-то не напечатал ни одного символа кроме цифр. Таким образом, номера i и n слились в единую последовательность цифр.

Теперь понять, какой же был номер у найденной страницы, стало большой проблемой, и решений у этой задачи может быть много. Роботу стало интересно, сколько существует решений, но так как робот не предназначен для решения таких задач, он нуждается в вашей помощи. Страницы в условиях нумеруются от **1** до **n** , числа i и n записываются без ведущих нулей.

Выясните, сколько есть корректных надписей вида «**Страница i из n** », при удалении из которых всех символов кроме цифр получается заданная во входном файле строка.

Входной файл содержит строку, состоящую только из цифр. Длина строки лежит в пределах от **1** до **200 000**, включительно.

Выведите количество корректных надписей вида «**Страница i из n** », при удалении из которых всех символов кроме цифр получается заданная во входном файле строка.

Решения:

Pascal ABC

```
program task_058;
var f,g: text;
    ch:char;
    a: array[1..200000] of char;
    i,k,t,s,p: Longint;
begin
Assign(f, 'input1.txt');
Assign(g, 'output.txt');
reset(f);
```

```

i:=0;
while not eof(f) do
begin
inc(i);
read(f,a[i]);
end;
close(f);
s:=0;
for k:=1 to i do
if a[k]='0' then s:=s+1;
s:=(i div 2)-s;
p:=(i div 2);
if i mod 2 = 0 then
for k:=1 to p do
if(strtoint(a[k])>strtoint(a[p+k]))
then begin s:=s-1; break; end;
rewrite(g);
write(g,s);
close(g);
end.

```

Задача №059. Шахмат

Напомним, что в шахматы играют два игрока, у каждого из которых изначально есть по 8 фигур и 8 пешек. В этой задаче пешки рассматривать не будем. Ни одна фигура, кроме коня, не может перепрыгивать через свои или чужие фигуры. Нельзя делать ход на поле, занятое своей фигурой. При ходе на поле, занятое чужой фигурой, она снимается с доски.

Фигуры ходят следующим образом:

- Король - на любую соседнюю по вертикали, горизонтали или диагонали клетку;
- Ферзь - на любое расстояние по вертикали, горизонтали или диагонали;

- Ладья - на любое расстояние по вертикали или горизонтали;
- Слон - на любое расстояние по диагонали;
- Конь - в форме буквы "Г": на 1 клетку по горизонтали и на 2 по вертикали, или наоборот, на 1 клетку по вертикали и 2 по горизонтали.

Вам даны позиции одной белой и одной чёрной фигуры. Определите, бьют ли эти фигуры друг друга, и если бьют, выведите какая из них бьёт какую.

Входные данные: Первая строка содержит тип и позицию белой фигуры, вторая строка - тип и позицию чёрной фигуры. Каждая фигура задаётся тремя символами, первый из которых обозначает тип, второй задаёт горизонталь от a до h, а третий задаёт вертикаль от 1 до 8. Типы фигур: B - слон, N - конь, R - ладья, Q - ферзь, K король. Гарантируется, что фигуры стоят на различных клетках доски.

Выходные данные: В выходной файл output.txt выведите одно слово - ответ на задачу.

- Если фигуры не бьют друг друга, выведите NONE
- Если обе фигуры бьют друг друга, выведите BOTH
- Если белая фигура бьёт чёрную, а чёрная не бьёт белую, выведите WHITE
- Если чёрная фигура бьёт белую, а белая не бьёт чёрную, выведите BLACK

№	input.txt	output.txt
1.	Ka1 Rg1	BLACK
2.	Qf3 Qh5	BOTH

Решения:

Pascal ABC

```
program task_059;
var s1,s2,f1,f2:string;
```

```

    n1,n2,m1,m2,c,k:integer;
    f,g:text;
begin
    Assign(f,'input-shahmat.txt');
    Assign(g,'output-shahmat.txt');
    Reset(f);
    Rewrite(g);
    Readln(f,s1);
    Readln(f,s2);
    close(f);
    f1:=s1[1];
    f2:=s2[1];
    n1:=ord(s1[2])-96;
    n2:=ord(s2[2])-96;
    m1:=ord(s1[3])-48;
    m2:=ord(s2[3])-48;
    if f1='B' then
        if (abs(n1-n2)=abs(m1-m2)) or
            (abs(n1-m2)=abs(m1-n2)) then c:=c+1;
    if f2='B' then
        if (abs(n1-n2)=abs(m1-m2)) or
            (abs(n1-m2)=abs(m1-n2)) then k:=k+1;
    if f1='N' then
        if ((abs(n1-n2)=2) and (abs(m1-m2)=1)) or
            ((abs(n1-n2)=1) and (abs(m1-m2)=2)) then c:=c+1;
    if f2='N' then
        if ((abs(n1-n2)=2) and (abs(m1-m2)=1)) or
            ((abs(n1-n2)=1) and (abs(m1-m2)=2)) then k:=k+1;
    if f1='R' then
        if (n1=n2) or (m1=m2) then c:=c+1;
    if f2='R' then
        if (n1=n2) or (m1=m2) then k:=k+1;
    if f1='Q' then
        if (abs(n1-n2)=abs(m1-m2)) or
            (abs(n1-m2)=abs(m1-n2)) or (n1=n2) or (m1=m2)

```

```

        then c:=c+1;
if f2='Q' then
    if (abs(n1-n2)=abs(m1-m2)) or
        (abs(n1-m2)=abs(m1-n2)) or
            (n1=n2)or(m1=m2) then k:=k+1;
if f1='K' then
    if ((n1=n2) and (abs(m1-m2)=1)) or
        ((abs(n1-n2)=1) and (m1=m2)) or
            ((abs(n1-n2)=1) and (abs(m1-m2)=1)) then c:=c+1;
if f1='K' then
    if ((n1=n2) and (abs(m1-m2)=1)) or
        ((abs(n1-n2)=1) and (m1=m2)) or
            ((abs(n1-n2)=1) and (abs(m1-m2)=1)) then k:=k+1;
if (k>0)and(c>0) then writeln(g,'BOTH')
else if c>0 then writeln(g,'WHITE')
else if k>0 then writeln(g,'BLACK')
else writeln(g,'NONE');
close(g);
end.

```

Задача №060. Призма

Имеется n – угольная призма (n – четное число), длины всех ребер (стороны оснований и высота призмы) равны **1**. Вершины занумерованы так:

Верхние вершины занумерованы от **1** до n по кольцу.

Под **1**-й вершиной $n+1$ – я, ... под n -й вершиной $2n$ – я.

По заданным двум различным натуральным числам, не превышающим $2n$, найти длину кратчайшего пути (по ребрам) между заданными вершинами.

Входные данные:

n – четное числа,

Два различные натуральные числа.

Выходные данные: Длина кратчайшего пути

№	<i>input.txt</i>	<i>output.txt</i>
1.	12 7, 14	6
2.	2012 3, 4000	28

Решения:

Pascal ABC

```

program task_060;
var a,b,n,k:integer;
    f,g:text;
begin
Assign(f,'input-prizma.txt');
Assign(g,'output-prizma.txt');
Reset(f);
Rewrite(g);
Readln(f, n);
Readln(f,a,b);
close(f);
if (a <= n) and (b <= n) then
    if abs(a-b)<(n div 2) then k:=abs(a-b)
        else k:=n-abs(a-b);

if (a > n) and (b > n) then
    if abs(a-b)<(n div 2) then k:=abs(a-b)
        else k:=n-abs(a-b);

if (a <= n) and (b > n) then
begin
a:=a+n;
if abs(a-b)<=(n div 2) then k:=abs(a-b)+1
    else k:=abs(n-abs(a-b))+1;
end;
end;
```

```

if (b <= n) and (a > n) then
begin
  b:=b+n;
  if abs(a-b)<=(n div 2) then k:=abs(a-b)+1
    else k:=abs(n-abs(a-b))+1;
end;
writeln(g,k);
close(g);
end.

```

Задача №061. Цифровая

По данному натуральному числу $1 < K < 2017$ найти количество K -значных чисел, произведение цифр которых равно 8.

Например: $K=2 \rightarrow 4$ [18, 24, 42, 81]

Ввод: одно натуральное число.

Вывод: одно натуральное число.

Решения:

Pascal ABC

Идея решения заключается в следующем.

- 1) Количество сочетаний с повторениями 8 умноженный на 1 вычисляется по следующей формуле $k!/(k-1)!*1!$;
- 2) Количество сочетаний с повторениями 2 умноженный на 4 вычисляется по следующей формуле $k!/(k-2)!*1!*1!$;
- 3) Количество сочетаний с повторениями умноженный $2*2*2$ вычисляется по следующей формуле $k!/(k-3)!*3!$.

Сложим эти формулы, упростим и получим общее количество по следующей формуле: $(k*(k+1)*(k+2))/6$.

```

program task_061;
var k:longint;
begin

```

```
read(k);
k:=k*(k+1)*(k+2);
writeln(k div 6);
end.
```

Задача №062. Выплата монетами

Показать, что любую сумму, большую 7 копеек, можно выплатить, используя только 3-х и 5-ти копеечные монеты. При выплате общая количество монет должен быть минимальным.

Решения:

<i>Pascal ABC</i>

```
program task_062;
uses crt;
var n:integer;
begin
  readln(n);
  case (n mod 5) of
    0: begin writeln(0); writeln(n/5); end;
    1: begin writeln(2);writeln((n-6) div 5); end;
    2: begin writeln(4);writeln((n-12) div 5); end;
    3: begin writeln(1);writeln((n-3) div 5); end;
    4: begin writeln(3);writeln((n-9) div 5); end;
  end;
end.
```

Задача №063. Длина отрезка

В плоскости находится $N > 2$ отрезок с координатами $(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)$. a_i – это начальная координата i ной отрезки, а b_i – конечная координата. Найти длины отрезка, который находятся во всей отрезке.

В первой строке задаются целое N – число отрезков. В следующих N строке записаны два целое число являющиеся координатами отрезков.

Нужно выводит одно целое число – длина отрезка находящее в пересечении всех отрезков. Выводит число – 0, если нет такой отрезок.

Решения:

Pascal ABC

```
program task_063;
var n,k,ai,bi,max,min: longint;
begin
  readln(n);
  max:=0; min:=2147483647;
  for k:=1 to n do
  begin
    readln(ai,bi);
    if max<ai then max:=ai;
    if min>bi then min:=bi;
  end;
  if max<min then writeln(min-max) else writeln(0);
end.
```

Задача №064. Шахматная доска

Координаты двух полей шахматной доски заданы в виде двух пар чисел x_1, y_1 и x_2, y_2 . На первом поле стоит **ферзь**, на втором – **конь**. Определить, **бьет ферзь коня, конь - ферзя**, или фигуры **не угрожают друг другу**.

В первой строке задаются координаты (x_1, y_1) ферзя, а во второй строке координаты (x_2, y_2) коня.

Нужно записать в выходной файл один из вариантов: «Ферзь бьет коня», «Конь бьет ферзя» или «Фигуры не угрожают друг другу».

Решения:**Pascal ABC**

```
program task_064;
var x1,y1,x2,y2:shortint;
begin
  readln(x1,y1);
  readln(x2,y2);
  if (x1=x2) or (y1=y2) or (abs(x1-x2)=abs(y1-y2))
  then writeln('Ферзь бьет коня')
  else if sqr(x1-x2)+sqr(y1-y2)=5
  then writeln('Конь бьет ферзя')
  else writeln('Фигуры не угрожают друг-другу');
end.
```

Задача №065. Кубы

Куб со сторонами **N** см. Раскрасили и разделили на кубы со сторонами **1 см**. В результате появились кубы с различным числом раскрашенных сторон. Составить программу нахождения числа кубов с различным числом раскрашенных сторон.

Входные данные: Одно **N** целое число – сторона первого куба. ($1 \leq N \leq 1000$)

Выходные данные: Вывести два целых числа – число раскрашенных сторон и число кубов, раскрашенных этим вариантом. Если есть куб, нераскрашенный ни одним из вариантов, тогда не выводить эту пару чисел.

Решения:**Pascal ABC**

```
program task_065;
var n:integer;
    b0,b1,b2,b3: shortint;
begin
  read(n);
```



```

b3:=8;
b2:=12*(n-2);
b1:=6*sqr(n-2);
b0:=(n-2)*sqr(n-2);
if b0<>0 then writeln('0 ', b0);
if b1<>0 then writeln('1 ', b1);
if b2<>0 then writeln('2 ', b2);
if b3<>0 then writeln('3 ', b3);
end.

```

Задача №066. Количество различных чисел и n^2

Дан массив **x: array[1..n] of integer**. Найти количество различных чисел среди элементов этого массива. Число действий должно быть порядка n^2 .

Решения:

Pascal ABC

```

program task_066;
const n=10;
var k,i,j: integer;
    x: array[1..n] of integer;
    y: boolean;
begin
for i:=1 to n do read(x[i]);
k:=0;
for i:=1 to n do
begin
j:=i+1;y:=true;
while (j<=n) and (y) do
if x[j]=x[i] then y:=false else j:=j+1;
if y then k:=k+1;
end;
write(k);
end.

```

Задача №067. Количество общих элементов

Даны два возрастающих массива из целых k и h элементов соответственно. Найти количество общих элементов в этих массивах, то есть количество тех элементов, для которых $x[i]=y[j]$ для некоторых i и j . Число действий порядка $k+h$.

Решения:

Pascal ABC

Возьмем дополнительные переменные $0 \leq k1 \leq k$ и $0 \leq h1 \leq h$. Искомое количество общих элементов будем хранить в n . Цикл повторяется $k+h$ раз. В теле цикла выполняется или одна, или три операции присваивания. В любом случае число действий порядка $k+h$.

```
program task_067;
const k=5; h=6;
var k1,h1,n,i: integer;
    x: array[1..k] of integer;
    y: array[1..h] of integer;
begin
  for i:=1 to k do readln(x[i]);
  for i:=1 to h do readln(y[i]);
  k1:=0; h1:=0; n:=0;
  while (k1<>k) and (h1<>h) do
    if x[k1+1]<y[h1+1]
      then k1:=k1+1
    else if x[k1+1]>y[h1+1]
      then h1:=h1+1
    else
      begin
        k1:=k1+1;
        h1:=h1+1;
      end
  end
end
```

```

        n:=n+1;
    end;
write(n);
end.

```

Задача №068. Соединение массива

Даны два массива $x[1] \leq \dots \leq x[k]$ и $y[1] \leq \dots \leq y[h]$. "Соединить" их в массив $z[1] \leq \dots \leq z[m]$ ($m=k+h$, каждый элемент должен входить в массив z столько раз, сколько раз он входит в общей сложности в массивы x и y). Число действий порядка m .

Решения:

Pascal ABC

Пусть у нас есть две стопки карточек, отсортированных по алфавиту. Мы соединяем их в одну стопку, выбирая каждый раз ту из верхних карточек обеих стопок, которая идет раньше в алфавитном порядке. Если в одной стопке карточки кончились, берем их из другой стопки.

```

program task_068;
const k=2; h=3;
var k1,h1,n,i: integer;
    x: array[1..k] of integer;
    y: array[1..h] of integer;
    z: array[1..k+h] of integer;
begin
for i:=1 to k do readln(x[i]);
for i:=1 to h do readln(y[i]);
k1:=0; h1:=0;
while (k1<>k) or (h1<>h) do
    if k1=k then begin
        h1:=h1+1;

```

```

        z[k1+h1]:=y[h1];
    end
else
    if h1=h then begin k1:=k1+1;
        z[k1+h1]:=x[k1] end
else
    if x[k1+1]<=y[h1+1] then begin k1:=k1+1;
        z[k1+h1]:=x[k1] end
        else
            if x[k1+1]>=y[h1+1] then begin h1:=h1+1;
                z[k1+h1]:=y[h1] end;
for i:=1 to k+h do write(z[i]);
end.

```

Задача №069. Наименьшее целое положительное

Дан неубывающий массив положительных целых чисел $a[1] \leq a[2] \leq \dots \leq a[n]$. Найти наименьшее целое положительное число, не представимое в виде суммы нескольких элементов этого массива (каждый элемент массива может быть использован не более одного раза). Число действий порядка n .

Решения:

<i>Pascal ABC</i>

Пусть известно, что числа, представимые в виде суммы элементов $a[1], \dots, a[k]$, заполняют отрезок от 1 до некоторого R . Если $a[k+1] > R+1$, то $R+1$ и будет минимальным числом, не представимым в виде суммы элементов массива. Если же $a[k+1] \leq R+1$, то числа, представимые в виде суммы элементов $a[1], \dots, a[k+1]$, заполняют отрезок от 1 до $R+a[k+1]$.

```

program task_069;
const n=5;

```

```

var i,k,r: integer;
  a: array[1..n] of integer;
begin
  for i:=1 to n do read(a[i]);
  k:=0;r:=0;
  while (k<>n) and (a[k+1]<=R+1) do
  begin
    r:=r+a[k+1];
    k:=k+1;
  end;
  writeln(r+1);
end.

```

Задача №070. Перестановка элементов массива

Дан массив **x[1..n]** и число **b**. Переставить числа в массиве таким образом, чтобы слева от некоторой границы стояли числа, меньшие или равные **b**, а справа от границы - большие или равные **b**.

Решения:

<i>Pascal ABC</i>

```

program task_070;
const n=5;
var i,b,h,r,m: integer;
  x: array[1..n] of integer;
begin
  readln(b);
  for i:=1 to n do
    read(x[i]);
  h:=0;r:=n;
  while h<>r do
  if x[h+1]<=b then h:=h+1
  else
    if x[r]>=b then r:=r-1

```

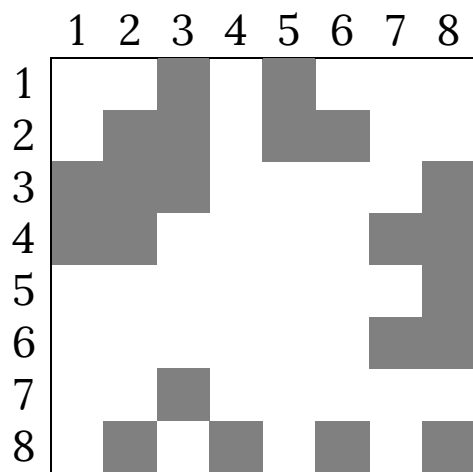
```

else begin
  m:=x[h+1]; x[h+1]:=x[r]; x[r]:=m;
  h:=h+1;r:=r-1;
end;
for i:=1 to n do writeln(x[i]);
end.

```

Задача №071. Прямоугольник максимальной площади

На квадратном клетчатом листе бумаги 8x8 клеток заштрихована часть клеток (пример на рисунке). Определить вписанный в решётку прямоугольник максимальной площади, не содержащий заштрихованных клеток. В качестве ответа вывести площадь прямоугольника и координаты его двух противоположных вершин. (Предполагается, что прямоугольник с максимальной площадью один.). Для приведенного примера координаты вершин (3,4) и (7,6), площадь 15 клеток.



Решения:

Pascal ABC

Представим лист бумаги в виде числовой матрицы A(8x8). Обозначим заштрихованные клетки единицами, а чистые - нулями. Данную программу удобно реализовать, используя подпрограммы. Напишем подпрограмму, которая проверяет, есть ли в прямоугольнике с координатами левой верхней вершины (i1,j1) и координатами правой нижней вершины (i2,j2) заштрихованные клетки (т.е. элементы, равные 1).

Создадим функцию которой будет возвращать значение "истина" (true), если заштрихованных клеток в рассматриваемом прямоугольнике нет, и "ложно" (false) - в противном случае.

В основной программе организуем два попарно вложенных цикла: в первом цикле будут изменяться координаты верхнего левого угла рассматриваемого прямоугольника, во втором - координаты нижнего правого угла. Если в прямоугольнике нет заштрихованных точек, то вычислим его площадь, и если она является максимальной, запоемним её и координаты противоположных вершин этого прямоугольника. Часть основной программы, реализующая данный алгоритм, будет иметь следующий вид:

```
program task_071;
const n=8;
type matr=array[1..n,1..n] of integer;
var i1,j1,i2,j2,s:integer;
    maxs,maxi1,maxj1,maxi2,maxj2:integer;
    a:matr;

function prov(a:matr;i1,j1,i2,j2:integer):boolean;
var i,j:integer;
begin
    prov:=true;
    for i:=i1 to i2 do
        for j:=j1 to j2 do
            if a[i,j]=1 then prov:=false;
        end;
    end;

begin
maxs:=0;
for i1:=1 to n do
for j1:=1 to n do
```

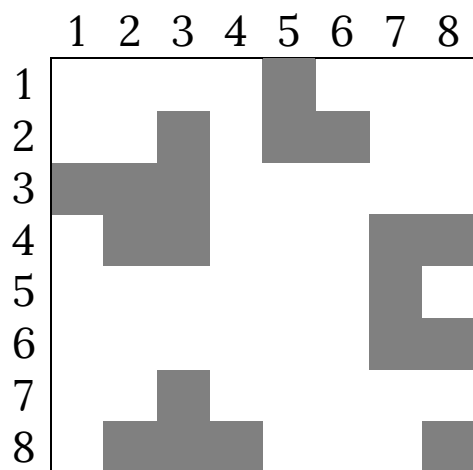
```

for i2:=i1 to n do
for j2:=j1 to n do
begin
s:=(i2-i1+1)*(j2-j1+1);
if prov(a,i1,j1,i2,j2)and(maxs<s) then
begin
maxs:=s;
maxi1:=i1;
maxj1:=j1;
maxi2:=i2;
maxj2:=j2;
end;
end;
writeln(' s=',maxs);
writeln('(',maxi1,',',maxj1,')', '(',maxi2,',',maxj2,')');
end.

```

Задача №072. Фигура

На квадратном клетчатом листе бумаги n на m клеток нарисовано несколько фигур, каждая из которых состоит только из целых клеток. Различные фигуры не накладываются и не соприкасаются (пример на рисунке). Определить фигуру максимальной площади. (В



качестве ответа вывести площадь фигуры и координаты одной из её точек. Предполагается, что фигура с максимальной площадью одна.) Например, на листе 8 на 8 клеток:

Решения:

Pascal ABC

Подобные задачи достаточно просто реализуются с использованием рекурсии. Решение построим следующим образом: представим лист в виде числовой матрицы $A(n \times m)$. Обозначим заштрихованные клетки единицами, а чистые - нулями. Напишем рекурсивную процедуру, которая определяет площадь заштрихованной фигуры и заменяет клетки уже просмотренной фигуры двойками (чтобы не просматривать эту фигуру ещё раз). В основной программе организуем цикл перебора всех элементов матрицы. Если очередной элемент равен 1 (ещё не просмотренная фигура), то будем вызывать процедуру определения площади этой фигуры).

Здесь n и m - число столбцов и строк в рассматриваемой матрице, i, j - координаты найденной клетки фигуры, s - переменная, в которой накапливается площадь фигуры. Процедура учитывает найденную клетку в площади, потом проверяет, заштрихована ли клетка, которая расположена ниже рассматриваемой (если это не последняя строка), и если да, то рекурсивно вызывается с этой клетки, потом тот же процесс повторяется для клеток расположенных справа, слева и сверху от рассматриваемой.

```
program task_072;
const n=8; m=8;
var a: array[1..n,1..m] of integer;
    i,j,im,jm,s,max:integer;

procedure zaliv(i,j:byte; var s:integer);
begin
    a[i,j]:=2;
    s:=s+1;
```

```

    if (i+1<=n)and(a[i+1,j]=1) then zaliv(i+1,j,s);
    if (j+1<=m)and(a[i,j+1]=1) then zaliv(i,j+1,s);
    if (j-1>0)and(a[i,j-1]=1) then zaliv(i,j-1,s);
    if (i-1>0)and(a[i-1,j]=1) then zaliv(i-1,j,s);
end;

begin
max:=0; im:=0; jm:=0;
for i:=1 to n do
  for j:=1 to m do read(a[i,j]);
for i:=1 to n do
for j:=1 to m do
  if a[i,j]=1 then
  begin
    s:=0;
    zaliv(i,j,s);
    if s>max then
    begin
      max:=s;
      im:=i;
      jm:=j;
    end;
  end;
writeln('Smax=',max,' im=',im,' jm=',jm);
end.

```

Задача №073. 8 ферзей

На шахматной доске необходимо расставить 8 ферзей так, чтобы они не угрожали друг другу.

Решения:

<i>Pascal ABC</i>

Ферзей 8, очевидно, что на каждой вертикали и горизонтали будет стоять по одному ферзю, поэтому

можно считать, что ферзь с номером k стоит на k-той вертикали и необходимо найти его координату по горизонтали.

```
program task_073;
  var n:longint;
      i1,i2,i3,i4,i5,i6,i7,i8 : byte;

function showstr(k :byte) : string;
var s : string;
begin
  s:='1';
  while length(s)<k do s:='0'+s;
  while length(s)<8 do s:=s+'0';
  showstr:=s;
end;

begin
  n:=0;
  for i1:=0 to 7 do
  for i2:=0 to 7 do
  for i3:=0 to 7 do
  for i4:=0 to 7 do
  for i5:=0 to 7 do
  for i6:=0 to 7 do
  for i7:=0 to 7 do
  for i8:=0 to 7 do
    if ((1 shl i1) or (1 shl i2) or (1 shl i3) or (1 shl i4)
      or (1 shl i5) or (1 shl i6) or (1 shl i7) or (1 shl i8))=255
      then n:=n+1;
  write(n);
end.
```

Задача №074. 4 ладьи

На шахматной доске размером **4x4** клетки расставить 4 ладьи так, чтобы они не угрожали друг другу. Определить все такие расстановки (всего их будет 24).

Решения:

Pascal ABC

Как и в предыдущей задаче, будем считать, что на каждой вертикали стоит по ладье, и для каждой из них необходимо найти координату по горизонтали (причем не совпадающую с соответствующими координатами остальных ладей). Таким образом, исходная задача сводится к нахождению всех возможных перестановок из 4 элементов. Известно, что число перестановок из n элементов равно $n! = 1 * 2 * 3 * \dots * n$. Например, из 3 элементов можно получить $3! = 1 * 2 * 3 = 6$ перестановок:

1 2 3

1 3 2

2 1 3

2 3 1

3 1 2

3 2 1

Так как алгоритмы перестановок часто используются в олимпиадных задачах, рассмотрим их подробнее.

Наиболее коротким и простым для запоминания является рекурсивный алгоритм получения перестановок. Пусть $X[n]$ - массив, элементы которого числа от 1 до n . Для упрощения программы будем использовать процедуру `printm`, печатающую массив X , и процедуру `swar(a,b)`, меняющую местами значения переменных a и b . Тогда программа рекурсивного получения перестановок (при $n=4$) будет иметь вид:

```

program task_074;
const n=4;
var x:array [1..n] of integer;
    i:integer;
procedure printm;
var i:integer;
begin
    for i:=1 to n do write(x[i], ' ');
    writeln;
end;
procedure swap(var a,b:integer);
var v:integer;
begin
v:=a; a:=b; b:=v
end;

procedure perest(k:integer);
var i:integer;
begin
    if k=n-1 then printm
    else
        for i:=k+1 to n do
            begin
                swap(x[k+1],x[i]);
                perest(k+1);
                swap(x[k+1],x[i]);
            end;
        end;
    begin
        for i:=1 to n do x[i]:=i;
        perest(0);
    end.

```

Задача №075. Арабские и римские цифры

Дано натуральное число N ($N < 500$) арабскими или римскими цифрами. Если число N дано арабскими цифрами, выразить число римскими цифрами, и если наоборот римскими, то выразить арабскими цифрами.

Входные данные:

N – натуральное число (арабскими или римскими цифрами)

Выходные данные:

Натуральное число (арабскими или римскими цифрами)

№	input.txt	output.txt
1.	28	XXVIII
2.	LXXII	72

Решения:

Pascal ABC

```
program task_075;
label 1,2,3;
var c,n,i:integer;w,n1:string;
    ch:char;
begin
    readln(n1);
    val(n1,n,c);
    if n<>0 then begin
1: if n>=100 then begin w:=w+'M';
n:=n-100; goto 1;end;
    if n>=50 then begin w:=w+'L';
n:=n-50;end;
2: if n>=10 then begin w:=w+'X';
n:=n-10;goto 2;end;
    if n=9 then begin w:=w+'IX';
n:=n-9;end;
```

```

if n>=5 then begin w:=w+'V';
n:=n-5;end;
if n=4 then begin w:=w+'IV';
n:=n-4; end;
3: if n>=1 then begin w:=w+'I';
n:=n-1;goto 3; end;
writeln(w);end
else begin n:=0;
for i:= 1 to length(n1) do
case n1[i] of
'M': n:=n+100;
'L': n:=n+50;
'X': n:=n+10;
'V': n:=n+5;
'T': n:=n+1;
end;
if (n1[i-1]='I')and (n1[i]='X') or (n1[i]='V')
then n:=n-2;
writeln(n);
end;
end.

```

Задача №076. Пополнение сосудов

Имеется по одному сосуда объемом **1, 2, 22, 23, ... ,2k** литров и **n** литров воды, где **k** Целое число ($0 \leq k \leq 16$). Необходимо разлить всю воду в минимальное количество сосудов. Сосуды должны быть заполнены полностью (**сосуд не может быть заполнен частично**). Гарантируется, что сосудов будет достаточно.

Входные данные. В единственной строке содержится целое число **n** ($1 \leq n \leq 65535$).

Выходные данные. Должны быть указаны объемы заполненных сосудов.

№	<i>input.txt</i>	<i>output.txt</i>
1.	201	1 8 64 128

Решения:

Pascal ABC

```

program task_076;
var n,k,i:integer;
    a:array[1..16]of integer;
begin
readln(n);
k:=16; i:=1;
while n>0 do begin
if n>=(1 shl k)then begin a[i]:= (1 shl k); i:=i+1; n:=n-(1 shl k); end;
k:=k-1;
end;
for k:=i-1 downto 1 do
writeln(a[k]);
end.

```

Задача №077. Карточки

Айбек и Гульзат играют в игру с карточками. Игра очень простая – есть колода карт, на каждой из которых написана буква латинского алфавита. Карты перемешиваются, выдаются участникам, и они составляют из них слова.

Айбек перемешивает колоду в несколько этапов. На каждом этапе сначала он берет по одной карте сверху колоды и раскладывает их на две стопки – сначала налево, потом направо, потом опять налево и т.д., пока колода не кончится. После этого он кладет левую стопку на правую. Эти действия повторяются к раз.

Гульзат знает, в каком порядке лежат карты в колоде в начале и как Айбек их перемешивает. Теперь она хочет узнать, как карты будут лежать после того, как Айбек их перемешивает. Помогите ей это сделать.

Входные данные. Первая строка входного файла **kartochki-inp.txt** содержит строку, описывающую состояние колоды до перемешивания. Строка состоит из заглавных латинских букв. **i**-я буква строки соответствует **i**-ой карте от низа колоды, т.е. первая буква – самая нижняя карта, последняя – самая верхняя. Длина строки не превышает **100** символов. Вторая строка содержит целое число $1 \leq k \leq 100$.

Выходные данные. В выходной файл **kartochki-out.txt** выведите состояние колоды после перемешивания в том же формате, что и во входном файле.

№	<i>input.txt</i>	<i>output.txt</i>
1.	ABCDEFGF	FDBGЕСА

Решения:

<i>Pascal ABC</i>

```

program task_077;
var f,g:text;
    st,st1,st2: string;
    k,i,j:shortint;
begin
assign(f,'kartochki-inp.txt');
assign(g,'kartochki-out.txt');
reset(f);
rewrite(g);
readln(f,st);
readln(f,k);
close(f);
for i:=1 to k do

```

```

begin
  st1:=""; st2:="";
  j:=1;
  while j<=length(st) do
  begin
    st1:=st[j]+st1;
    j:=j+1;
    if j>length(st) then break;
    st2:=st[j]+st2;
    j:=j+1;
  end;
  st:=st2+st1;
end;
writeln(g,st);
close(g);
end.

```

Задача №078. Сумма длинных чисел

Даны два целых неотрицательных длинных числа **M** и **N**. Найти их сумму и вывести сколько значное получилось число, а также остаток от деления суммы на 10^{15} . Ограничения: $0 < M, N < 10^{30}$.

Входные данные. Ввод из файла **in_r78.txt** В первой строке содержится **M**, во второй - **N**.

Выходные данные. Вывод в файл **out_r78.txt** В первой строке вывести сколько значным числом получилась сумма, во второй строке вывести остаток от деления суммы **N+M** на 10^{15} без пробелов и ведущих нулей.

№	input.txt	output.txt
1.	12345678901234567890123456789 11111111111111111111111111111111	29 679001234567900

Решения:

Pascal ABC

```
program task_078;
const n=30;
var st1,st2: string[n];
    a,b,c:array[1..n] of integer;
    i,k,memo:integer;
    bool:boolean;
    f,g:text;
begin
assign(f,'in_r78.txt');
assign(g,'out_r78.txt');
reset(f);
rewrite(g);
readln(f,st1);
read(f,st2);
k:=length(st1);
for i:=n downto n+1-k do a[i]:=strtoint(st1[(i-(n-k))]);
k:=length(st2);
for i:=n downto n+1-k do b[i]:=strtoint(st2[(i-(n-k))]);
memo:=0;
for i:=n downto 1 do
begin
k:= a[i]+b[i]+memo;
if k>=10 then begin c[i]:=k mod 10; memo:=k div 10; end
else begin c[i]:=k; memo:=0; end;
end;
for i:=1 to n do
if c[i]<>0 then begin k:=i; break; end;
writeln(g,n-k+1);
for i:=n-k-12 to n do write(g,c[i]);
close(f);
close(g);
end.
```

Задача №079. Персиковое дерево

В саду бабушки Айжан растут различные плодовые деревья. В день рождения Айжан бабушка посадила одно персиковое дерево и сказала, чтобы она съедала каждый поспевший плод. В этом году дерево впервые дало плоды. Айжан сосчитала на дереве N -ое число плодов. Когда созрели первые плоды Айжан сосчитала их и съела. Они были красивыми, большими, сочными плодами числа K . Каждый день плодов созревало на число M больше, чем в предыдущий день. Бабушка говорила Айжан, что персики ей очень полезны и потому она должна каждый день побольше их есть. На счастье, персики были очень вкусными, поэтому она радовала бабушку съедая созревавшие каждый день персики. Сколько персиков съест Айжан в последний день, когда созреет N -ый персик?

Входные данные. Заданы числа N, K, M . Все числа не превышают 1000000000.

Исходные данные. Вывести одно целое число, являющееся ответом задания.

№	input.txt	output.txt
1.	10 1 1	4

Решения:

Pascal ABC

```
program task_079;
var f,g:text;
    n,k,m,t:ShortInt;
begin
  assign(f,'shabdaly-inp.txt');
  assign(g,'shabdaly-out.txt');
  reset(f); rewrite(g);
```

```

readln(f,n,k,m);
t:=k;
while t<=n do
begin
k:=k+m;
t:=t+k;
end;
if (t-k)=n then writeln(g,k-m)
else writeln(g,n-(t-k));
close(f);close(g);
end.

```

Задача №080. Генератор комбинаций

Напишите программу, которая принимает 2 числа **n** и **k** и выводит все битовые комбинации длины **n** с **k**-единицами в убывающем порядке (когда битовые комбинации рассматриваются как двоичные числа). Можете предполагать, что $30 \geq n > 0$ и $8 > k \geq 0$, и $n \geq k$. Строка ввода – это количество пар **n** и **k**. Числа **n** и **k** разделяются одиночным пробелом. Нули в начале битовой комбинации должны быть включены.

№	input.txt	output.txt
1.	2 1	10 01
2.	0 1	00
3.	2 0	1100 1010 1001 0110 0101 0011

Решения:

Pascal ABC

```
program task_080;
var f,g:text;
    s:string;
    i,n,k,p:integer;
begin
  assign(f,'in_r80.txt');
  assign(g,'out_r80.txt');
  reset(f);
  rewrite(g);
  read(f,n,k);
  s:="";
  for i:=1 to n do
    if i<=k then s:=s+'1' else s:=s+'0';
  writeln(g,s);
  p:=pos('10',s);
  while p>0 do
    begin
      s[p]:='0';
      s[p+1]:='1';
      writeln(g,s);
      p:=pos('10',s);
    end;
  close(f);
  close(g);
end.
```

Задача №081. Количество островов

Каждый элемент квадратной матрицы размерности $N \times N$ равен нулю, либо единице. Найдите количество «островов», образованных единицами. Под «островом» понимается группа единиц (либо одна единица), со всех сторон окруженная нулями (или краями

матрицы). Единицы относятся к одному «острову», если из одной из них можно перейти к другой «наступая» на единицы, расположенные в соседних клетках. Соседними являются клетки, граничащие по горизонтали или вертикали.

Входные данные. В первой строке входного файла **in_r81.txt** записано натуральное число **N** ($3 \leq N \leq 100$) – размер квадратной матрицы. В следующих **N** строках задаются элементы матрицы через пробел.

Выходные данные. В выходной файл **out_r81.txt** выведите единственное число – количество островов.

№	<i>input.txt</i>	<i>output.txt</i>
1.	5 1 0 1 1 1 0 0 0 0 0 1 1 1 0 1 0 1 0 0 1 0 0 0 1 1	4

Решения:

Pascal ABC

```

program task_081;
const n=101;
var mas:array[1..n,1..n] of integer;
    i,j,k,res: integer;
    f,g:text;

procedure count(i,j:integer);
begin
if mas[i,j]<>1 then exit;
mas[i,j]:=0;
count(i+1,j);
count(i-1,j);
count(i,j+1);

```

```

    count(i,j-1);
end;
begin
assign(f,'in_r81.txt');
assign(g,'out_r81.txt');
reset(f);
rewrite(g);
read(f,k);
for i:=2 to k+1 do
  for j:=2 to k+1 do
    read(f,mas[i,j]);
  for i := 1 to n do
    for j := 1 to n do
      begin
        if mas[i,j]=1 then inc(res);
        count(i,j);
      end;
    writeln(g,res);
  close(f);
  close(g);
end.

```

Задача №082. Произведение длинных чисел

Даны длинные целые неотрицательные числа **M** и **N**. Найти произведение этих чисел **M*N** и вывести сколько значное получилось число, а также остаток от деления результата произведения на 10^{15} . Ограничения: $0 < M, N < 10^{2500}$.

Входные данные. Ввод из файла **in_r82.txt**. В первой строке находится число **M**, во второй – **N**.

Выходные данные. Вывод в файл **out_r82.txt**. В первой строке вывести сколько значным числом получился результат произведения, а во второй строке

вывести остаток от деления результата произведения на 10^{15} без ведущих нулей и пробелов.

№	<i>input.txt</i>	<i>output.txt</i>
1.	9876543210 1023456789	20 215200126352690

Решения:

Pascal ABC

```
program task_082;
var
  buf:integer; //полученное число, в результате умножения
  des:integer; //счетчик "десяток"
  n,m:integer; //счетчики длины строки
  i,j,q:integer; //переменные для работы со строкой и массивами
  x1,x2:string; //сюда вводим числа в виде строк
  c,d:array[1..50] of integer; //массивы для хранения и оперирования
  с числами
  res:array[1..2500] of integer; //результат вычислений
  f,g:text;

begin
  assign(f,'in_r82.txt');
  assign(g,'out_r82.txt');
  reset(f);
  rewrite(g);
  readln(f,x1);
  n:=length(x1);
  readln(f,x2);
  m:=length(x2);

  //перенос первого числа в массив
  i:=1;
  repeat
    val(x1[i],c[i],q);
    i:=succ(i);
```

```

until i>n;
//перенос второго числа в массив
i:=1;
repeat
val(x2[i],d[i],q);
i:=succ(i);
until i>m;

//умножение двух чисел (умножение "столбиком")
for i:=length(x1) downto 1 do
begin
for j:=length(x2) downto 1 do
begin
buf:=c[i]*d[j]+des;
if buf>9 then
begin
res[i+j-1]:=buf mod 10;
des:=buf div 10;
end
else
begin
res[i+j-1]:=buf;
des:=0;
end;
end;
end;
end;

//вывод полученного числа
writeln(g,length(x1)+length(x2));
if res[1]=0 then
for i:=3 to length(x1)-1+length(x2) do write(g,res[i])
else
for i:=2 to length(x1)-1+length(x2) do write(g,res[i]);
close(f);
close(g);
end.

```

Задача №083. Сумма всех чисел

Найдите сумму всех чисел, поданных на вход вашей программе.

Входные данные. На вход подается не более **1000** целых чисел, не превосходящих **1000** по абсолютной величине. Числа разделены пробелами и/или переводами строк.

Выходные данные. Одно число – сумма.

№	input.txt	output.txt
1.	1000 1000 -1000 1	1001

Решения:

Pascal ABC

```
program task_083;
var a,s:longint;
    f,g:text;
begin
  assign(f,'in_r83.txt');
  assign(g,'out_r83.txt');
  reset(f);
  rewrite(g);
  s:=0;
  while not eof(f) do
  begin
    read(f,a);
    s:=s+a;
  end;
  write(g,s);
  close(f);
  close(g);
end.
```

Задача №084. Последовательность

Заданы два первых члена арифметической прогрессии (арифметической прогрессией называется последовательность чисел a_1, a_2, \dots, a_n , такая что $a_i - a_{i-1} = \text{const}$ для всех $i \geq 2$). Найдите ее N -й член.

Входные данные. В строке содержится три числа - a_1 , a_2 и N ($1 \leq a_1 \leq a_2 \leq 1000$, $3 \leq N \leq 1000$).

Выходные данные. Требуется вывести a_n .

№	input.txt	output.txt
1.	3 4 5	7

Решения:

Pascal ABC

```
program task_084;
var a1,a2,n:longint;
begin
  read(a1,a2,n);
  write(a1+(n-1)*(a2-a1))
end.
```

Задача №085. Большие числа

Семисегментный индикатор используется для отображения цифр путем включения определенных комбинаций сегментов:



У вас есть неограниченное количество таких индикаторов, но мощность батареи хватит только на то,

чтобы включить N сегментов. Найдите наибольшее число, которое вы сможете отобразить.

Входные данные. В единственной строке записано целое число $2 \leq N \leq 100$).

Выходные данные. Выведите наибольшее число которое можно отобразить на индикаторе включив N сегментов.

№	input.txt	output.txt
1.	4	11

Решения:

Pascal ABC

```
program task_085;
var n,i:byte;
begin
  read(n);
  if n mod 2=0 then for i:=1 to n div 2 do write(1)
  else
    begin
      write(7);
      n:=n-3;
      for i:=1 to n div 2 do write(1);
    end;
end.
```

Задача №086. Allies

У Миши и Толи **M** и **N** друзей соответственно. Из них **K** друзей знают Мишу и Толю одновременно.

Помогите узнать, сколько друзей знают или Мишу, или Толю, но не обоих вместе.

Входные данные. Первая строка входного файла содержит одно целое число **T** — количество тестов

($1 \leq T \leq 104$). В каждой из следующих **T** строк содержится три целых числа **M, N, K** ($0 \leq M, N \leq 10^9, 0 \leq K \leq \min(M, N)$).

Выходные данные. Для каждого теста в отдельной строке необходимо вывести одно число – количество друзей, которые знают или Мишу, или Толю, но не обоих вместе.

№	input.txt	output.txt
1.	3 3 2 1 20 23 15 30 30 30	3 13 0

Решения:

Pascal ABC

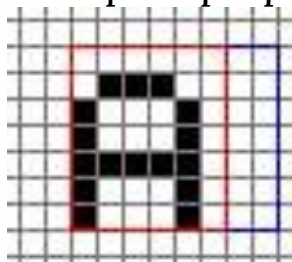
```

program task_086;
var t,m,n,k,i:longint;
begin
  readln(t);
  for i:=1 to t do
  begin
    readln(m,n,k);
    writeln(m+n-2*k);
  end;
end.
```

Задача №087. Нарисовать Символ

Ввод текста на экран – необходимая операция. Это самый распространённый способ отображения информации. Для вывода текста должным образом, нужны символы. Допустим, наш компьютер может отображать только символы определённого размера – 8×8 , 8×14 и 8×16 пикселей. Символы подобны обычным изображениям или спрайтам. Основное отличие

заключается в том, что в данном случае мы не должны хранить информацию о цвете для каждого пикселя. Это означает, что нам не нужны байты для хранения отдельных пикселей, а достаточно битов. Пример битовой карты буквы "А", при размере шрифта 8×8:



Каждую строку можно представить 8 битами, таким образом, символ 8×8 требует 8 байт. Определение битовой карты достаточно просто: **1** - где надо отобразить пиксель, **0** - в противном случае. Каждая строка пикселей хранится в одном байте. Пример хранения буквы «А» в различных системах счисления по строкам:

Двоичное представление	Шестнадцатеричное представление	Десятичное представление
00000000	00	0
01110000	70	112
10001000	88	136
10001000	88	136
11111000	F8	248
10001000	88	136
10001000	88	136
00000000	00	0

Требуется написать программу, которая по заданному рисунку составит битовую карту символа.

Входные данные. Строка содержит три целых числа **N**, **M** и **K**. **N** – это количество символов, от **1** до **255**), **M** – ширина, всегда равна **8**, **K** – высота, может быть **8**, **14** или **16**. Далее идут изображения **N** – символов. Каждый символ

представлен **K** строками и **M** столбцами. Где «.» - обозначает пустой пиксель, «*» - обозначает заполненный пиксель.

Выходные данные. Вывести **N** строк, которые описывают в десятичной системе каждый из **N** символов. В каждой строке, через пробел должны быть указаны **K** - целых десятичных числа.

№	<i>input.txt</i>	<i>output.txt</i>
1.	2 8 8 ***.... *...*... *...*... *****... *...*... *...*...	0 0 0 0 0 0 0 0 0 112 136 136 248 136 136 0

Решения:

Pascal ABC

```

program task_087;
var n,m,k,i,j,i1,s1,s:byte;
    ch:char;
    ss1,ss:string;
    ss2:array[1..255] of string;
begin
  readln(n,m,k);
  for i:=1 to n do begin

```



```

ss:="";
begin
  for j:=1 to k do
    begin
      s:=128;s1:=0;
      for i1:=1 to m do
        begin
          read(ch);
          if ch='*' then s1:=s1+s;
          s:=s div 2;
        end;
        str(s1,ss1);
        ss:=ss+' '+ss1;
        readln;
      end;
    end;
  delete(ss,1,1);
  ss2[i]:=ss;
end;
for i:=1 to n do
  writeln(ss2[i]);
end.

```

Задача №088. Маршруты

В Бишкеке есть N школ, от каждой школы выходит отдельная дорога к другим $N-1$ школам. Требуется написать программу, которая посчитает количество уникальных дорог между школами.

Входные данные. Строка содержит одно целое число N от 2 до 2^{16} .

Выходные данные. Вывести одно число – количество уникальных дорог между школами.

№	<i>input.txt</i>	<i>output.txt</i>
1.	5	10

Решения:

Pascal ABC

```

program task_088;
  var n,k:longint;
begin
  read(n);
  if n=2 then write(1)
  else
if odd(n) then write(((n-1) div 2)*n) else write((n div 2)*(n-1));
end.

```

Задача №089. Треугольник

Даны три меры угла в градусах, требуется определить – могут ли эти три угла являться углами одного треугольника.

Входные данные. Строка содержит три целых числа **A, B, C** (числа от 0 до 180).

Выходные данные. Вывести **“YES”** – если могут являться сторонами треугольника, **“NO”** – в противном случае.

№	<i>input.txt</i>	<i>output.txt</i>
1.	80 10 90	YES
2.	90 90 0	NO

Решения:

Pascal ABC

```

program task_089;
  var a,b,c:integer;
begin

```

```

read(a,b,c);
if (a+b+c=180) and (a>0) and (b>0) and (c>0) then
write('YES') else write('NO')
end.

```

Задача №090. Прогрессия

Дана последовательность из **N** чисел. Требуется определить, являются ли они арифметической прогрессией в том порядке, в котором они расположены.

Входные данные. Первая строка содержит число **N** (от 3 до 1000). В следующей строке находятся **N** целых чисел (каждое от -10^9 до 10^9)

Выходные данные. Вывести “**YES**” – если числа являются арифметической прогрессией, “**NO**” – в противном случае.

№	<i>input.txt</i>	<i>output.txt</i>
1.	3 1 2 3	YES
2.	4 1 3 2 4	NO

Решения:

Pascal ABC

```

program task_090;
var n,i:integer;
    d:longint;
    f:boolean;
    a:array[1..10000] of longint;
begin
  readln(n);
  for i:=1 to n do
    read(a[i]);

```

```

d:=a[2]-a[1];
f:=true;
for i:=3 to n do
  if d<>a[i]-a[i-1] then begin f:=false;break;end;
  if f then write('YES') else write('NO');
end.

```

Задача №091. Многоугольник

Дан многоугольник, определить, сумму всех его внутренних углов.

Входные данные. N – количество углов многоугольника, от 3 до 1 000 000.

Выходные данные. Единственное число – сумму всех углов многоугольника.

№	<i>input.txt</i>	<i>output.txt</i>
1.	3	180

Решения:

<i>Pascal ABC</i>

```

program task_091;
var n:longint;
begin
  read(n);
  write(180*(n-2))
end.

```

Задача №092. День рождения

Петя знает день рождения своего Босса. Но он не знает день недели этой даты в текущем **2016** году.

Входные данные. На вход подается два целых числа **d**, **m**, где день рождения Босса – **d** ($1 \leq d \leq 31$), месяц рождения – **m** ($1 \leq m \leq 12$).

Выходные данные. Вывести строку, содержащую название дня недели, на которой приходится день рождения Босса в текущем году. Названия дней недели: “Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”, “Saturday”, “Sunday”.

№	<i>input.txt</i>	<i>output.txt</i>
1.	30 4	Saturday

Решения:

Pascal ABC

```
program task_092;
var d,m,i,k:integer;
    a:array[0..11] of integer;
begin
  readln(d,m);
  a[0]:=0;
  a[1]:=31;a[2]:=60;a[3]:=91;a[4]:=a[3]+30;a[5]:=a[4]+31;
  a[6]:=a[5]+30;a[7]:=a[6]+31;a[8]:=a[7]+31;a[9]:=a[8]+30;
  a[10]:=a[9]+31;a[11]:=a[10]+30;
  k:=d+a[m-1];
  k:=k mod 7;
  case k of
    0: write('Thursday');
    1: write('Friday');
    2: write('Saturday');
    3: write('Sunday');
    4: write('Monday');
    5: write('Tuesday');
    6: write('Wednesday');
  end;
end;
```

Задача №093. N мер углов

Даны **N** мер углов в градусах, требуется определить – могут ли эти углы являться углами одного выпуклого многоугольника с **N** углами.

Входные данные. Первая строка содержит число **N** ($3 \leq N \leq 360$). Далее идут **N** целых чисел (числа от 0 до 180).

Выходные данные. В первой строке вывести сумму мер углов многоугольника. Во второй строке вывести “**YES**” – если все углы могут являться углами многоугольника, “**NO**” – в противном случае.

№	<i>input.txt</i>	<i>output.txt</i>
1.	3 80 10 90	180 YES
2.	4 90 90 90 0	270 NO

Решения:

Pascal ABC

```
program task_093;
  var s,n,a,i:longint;
begin
  s:=0;
  readln(n);
  for i:=1 to n do
  begin
    read(a);
    s:=s+a;
  end;
  writeln(s);
  if (n-2)*180=s then write('YES') else write('NO');
end.
```

Задача №094. Остатки

Пусть дано N целых чисел и целое число A . При делении этих чисел на A получается новый набор чисел – остатки от деления каждого числа на A . Например, у нас есть числа 1, 3, 4 и число 2. После деления на 2 у нас получаются остатки 1, 1, 0. Требуется написать программу, которая находит сумму различных чисел в наборе остатков после деления на A и проверяет, превышает ли она A .

Входные данные. В первой строке содержится два числа N и A . Где $1 \leq N \leq 10000$ и $1 < A < 2^{31}$. В следующей строке содержится N целых чисел, где каждое число лежит в диапазоне от 1 до 2^{31} .

Выходные данные. Вывести сумму различных чисел в наборе в первой строке. Во второй строке ответ “YES” – если сумма больше числа A и “NO” – если сумма меньше или равна A .

№	input.txt	output.txt
1.	10 11 1 2 3 4 5 6 7 8 9 10	55 YES
2.	10 17 17 34 51 68 85 102 119 136 153 170	0 NO

Решения:

Pascal ABC

```
program task_094;
var s,n,a,i,c,m,k,j:longint;
    b:array[1..100000] of longint;
    f:boolean;
begin
  k:=1;
  readln(n,a);
```

```

for i:=1 to n do
  begin
    read(c);
    b[k]:=c mod a;
    k:=k+1;
  end;
s:=b[1];
for i:=2 to n do
  begin
    f:=true;
    for j:=1 to i-1 do
      if b[i]=b[j] then begin f:=false;break;end;
      if f then s:=s+b[i];
    end;
  writeln(s);
  if s>a then write('YES') else write('NO');
end.;

```

Задача №095. Снова многоугольник

Задано натуральное число N от 4 до 10^6 . Нарисовать на клетчатой бумаге многоугольник периметра N , стороны которого идут по сторонам клеток, с наибольшей возможной площадью S , и вывести число – остаток от деления S на **1000**. Если таких многоугольников не существует, то вывести (-1).

Входные данные. Одно натуральное число N ($4 \leq N \leq 10^6$).

Выходные данные. Одно целое число – остаток от деления S на 1000.

№	<i>input.txt</i>	<i>output.txt</i>
1.	5	-1
2.	8	4

Решения:

Pascal ABC

```
program task_095;
var k,n:longint;
begin
  read(n);
  k:=n mod 4;
  if k=0 then write(((n div 4)*(n div 4)) mod 1000)
    else if (k=1) or (k=3) then write(-1)
    else
      begin
        n:=n div 2;
        write(((n div 2)*(n div 2+1)) mod 1000)
      end;
end.
```

Задача №096. Загрузка файлов

В корпорации **USRK** есть отдел разработок системного программного обеспечения. Стажеру Коле сказали написать модуль операционной системы, который должен загружать определенные файлы в память компьютера. Память компьютера линейная и адреса начинаются с **0**. Но есть некоторые ограничения на загрузку – процессор работает с файлами, загруженными только по определенным адресам, кратным 16 байтам.

Есть список файлов с их названиями и размером в байтах. Гарантировано, все файлы могут быть загружены в память. Надо помочь стажеру Коле написать программу, которая укажет, по каким адресам загружать файлы, так что бы они все находились по адресам кратным **16** и при этом адрес последнего файла был минимальным. Файлы должны быть загружены в память в том порядке, в котором их дают в исходном списке.

Входные данные. В первой строке находится целое число **N** – количество файлов для загрузки ($1 < N \leq 10^5$). Далее идут **N** строк, которые содержат имя файла и через пробел его размер в байтах – целое число **B** ($0 < B \leq 10^9$). Имя файла может состоять из маленьких и больших букв латинского алфавита, цифр и точки – «.». Длина имени файла может быть максимально 1000 символов. Имена файлов в списке не могут повторяться.

Выходные данные. Вывести **N** строк, где в каждой строке содержится адрес (начиная с 0) и через пробел имя файла. В **N+1** строке вынести первый свободный адрес кратный 16 байтам после загрузки файлов.

№	<i>input.txt</i>	<i>output.txt</i>
1.	4 IO.SYS 10 MSDOS.SYS 20 CONFIG.SYS 30 AUTOEXEC.BAT 40	0 IO.SYS 10 16 MSDOS.SYS 48 CONFIG.SYS 80 AUTOEXEC.BAT 128

Решения:

Pascal ABC

```

program task_096;
var n,i,x,p:longint;
    t,t1:longint;
    code:integer;
    s,s1:string;
begin
  readln(n);
  t:=0;
  for i:=1 to n do
  begin
    readln(s);
    p:=pos(' ',s);
    s1:=copy(s,1,p-1);
  
```

```

delete(s,1,p);
val(s,x,code);
writeln(t*16,' ',s1);
t1:=x div 16;
if x mod 16<>0 then t1:=t1+1;
t:=t1+t;
end;
write(t*16);
end.

```

Задача №097. Уравнение

По заданным целым числам **A, B, C** найти количество решений уравнения

$$(X^2+A)(X^2+B)(X+C)=0$$

в вещественных числах.

Входные данные. Строка с тремя целыми числами A, B, C (-100<A,B,C<100), разделенными одинарными пробелами.

Выходные данные. Одно целое неотрицательное число – количество решений уравнения в вещественных числах.

№	<i>input.txt</i>	<i>output.txt</i>
1.	5 5 2	1
2.	-81 -11 -9	4

Решения:

Pascal ABC

```

program task_097;
var a,b,c,t:integer;
begin
read(a,b,c);

```

```

if a>b then begin t:=a;a:=b;b:=t;end;
if (b<0) and (a<>b) and (c*c<>-b) and (c*c<>-a) then write(5);
if (b<0) and (a<>b) and ((c*c=-b) or (c*c=-a)) then write(4);
if (b<0) and (a=b) and (-a<>c*c) then write(3);
if (b<0) and (a=b) and (c*c=-a) then write(2);
if (b=0) and (a<>b) and (c*c<>-b) and (c*c<>-a) then write(4);
if (b=0) and (a<>b) and ((c*c=-b) or (c*c=-a)) then write(3);
if (b=0) and (a=b) and (c=0) then write(1);
if (b=0) and (a=b) and (c<>0) then write(2);
if (b>0) and (a<0) and (c*c<>-a) then write(3);
if (b>0) and (a<0) and (c*c=-a) then write(2);
if (b>0) and (a=0) and (c*c<>-a) then write(2);
if (b>0) and (a=0) and (c=0) then write(1);
if a>0 then write(1);
end.

```

Задача №098. Арифметическая последовательность

Ваша задача состоит в том, чтобы найти в заданной последовательности арифметическую подпоследовательность (идущих подряд чисел) максимальной длины.

Напомним: числа a_i, a_{i+1}, \dots, a_m образуют арифметическую последовательность с разностью M , если для каждого целого j такого, что $i \leq j < m$ верно равенство $a_j + M = a_{j+1}$. При этом, число $m - j + 1$ называется длиной последовательности.

Входные данные. В первой строке дано число N ($2 \leq N \leq 10^5$), количество целых чисел. Далее, через пробел или перевод строки дано N целых чисел в диапазоне $[-2768, 32767]$.

Выходные данные. На выходе должно быть выведено ровно одно число – длина максимальной арифметической последовательности.

№	<i>input.txt</i>	<i>output.txt</i>
1.	11 1 3 5 7 6 5 4 3 2 1 0	8

Решения:

Pascal ABC

```

program task_098;
var n,i,k,max,d:longint;
    a:array[1..100000] of integer;
begin
  readln(n);
  read(a[1]);
  read(a[2]);
  d:=a[2]-a[1];
  k:=2;max:=1;
  for i:=3 to n do
    begin
      read(a[i]);
      if d=a[i]-a[i-1] then inc(k)
      else
        begin
          if k>max then max:=k;
          k:=2;
          d:=a[i]-a[i-1];
        end;
    end;
  if k>max then max:=k;
  write(max);
end.

```

Задача №099. Правильная дробь

Правильная дробь – это дробь, в которой числитель меньше знаменателя. Требуется написать программу, которая определит, правильная или неправильная дробь дана.

Входные данные. В первой строке дан числитель N , во второй знаменатель D , где $0 < N, D \leq 10^{10000}$.

Выходные данные. Вывести “TRUE”, если дробь правильная и “FALSE” в противном случае.

№	<i>input.txt</i>	<i>output.txt</i>
1.	1 3	TRUE
2.	4 3	FALSE

Решения:

Pascal ABC

```
program task_099;
var n,d:string;
    l1,l2:integer;
begin
  readln(n);
  readln(d);
  l1:=length(n);
  l2:=length(d);
  if (l1<l2) or ((l1=l2) and (n<d))
    then write('TRUE')
    else write('FALSE');
end.
```

Задача №100. RLE

Был текст, состоящий из заглавных букв латинского алфавита, к нему применили алгоритм RLE (заменяли подряд идущие повторяющиеся буквы на количество повторяющихся букв и саму букву, если буква одна, то количество не пишется). Требуется восстановить исходный текст.

Входные данные. Строка, к которой применили алгоритм **RLE**. Исходная строка не может быть больше 10000 символов.

Выходные данные. Вывести исходную строку.

№	<i>input.txt</i>	<i>output.txt</i>
1.	B2C11A3BD	BCCAAAAAAAAAAAAABBBD

Решения:

Pascal ABC

```
program task_100;
var s,s1:string;
    n,i,j,code:integer;
begin
n:=1;
read(s);s1:="";
for i:=1 to length(s) do
begin
if (s[i]>='0') and (s[i]<='9') then s1:=s1+s[i]
else
begin
if s1="" then n:=1
else val(s1,n,code);
for j:=1 to n do write(s[i]);
s1:="";
end
end
end
```

```
    end;  
end;  
end.
```

Задача №101. Сумма делителей

Найдите сумму всех натуральных делителей заданного числа **N**.

Входные данные. В строке содержится число **N** ($1 \leq N \leq 1000$).

Выходные данные. Вывести искомую сумму.

№	<i>input.txt</i>	<i>output.txt</i>
1.	5	6
2.	10	18

Решения:

Pascal ABC

```
program task_101;  
var n,s,i:integer;  
begin  
  read(n);  
  s:=1;  
  for i:=2 to n div 2 do  
    if n mod i=0 then s:=s+i;  
  write(s+n);  
end.
```

Задача №102. Обратная сумма

Для каждого натурального числа **X** можно определить "развёрнутое" число **Rev(X)** следующим образом:

1. Число **X** записывается в десятичной системе счисления, например: 1021400.
2. Первая цифра меняется местами с последней, вторая – с предпоследней, и так далее. В результате цифры в строке оказываются в обратном порядке, например: 0041201
3. Все нули, идущие в начале строки, стираются, например: 41201.
4. Получившееся число называется $Rev(X)$.

Например, если $X=123$ то $Rev(X)=321$ и если $X=100$ то $Rev(X)=1$.

Вам дано два целых чисел **A** и **B** подсчитайте **$Rev(Rev(A)+Rev(B))$** .

Входные данные. В единственной строке задано два целых чисел A и B ($1 \leq A, B \leq 10^3$).

Выходные данные. В единственной строке выведите ответ задачи.

№	<i>input.txt</i>	<i>output.txt</i>
1.	123 100	223
2.	111 111	222
3.	5 5	1
4.	1000 1	2
5.	456 789	1461

Пояснение к первому тесту:

$$Rev(123) = 321, Rev(100) = 1, Rev(321 + 1) = Rev(322) = 223$$

Решения:

Pascal ABC

```
program task_102;
  var a,b:integer;

function rev(n:integer):integer;
  var s:integer;
begin
  s:=0;
  while n>0 do
  begin
    s:=s*10+n mod 10;
    n:=n div 10;
  end;
  rev:=s;
end;

begin
  readln(a);
  readln(b);
  write(rev(rev(a)+rev(b)));
end.;
```

Задача №103. Футбольная лига

Как вы знаете, в футбольном матче победитель получает 3 очка, а проигравший 0 очков. В случае ничьи, обе команды получают по 1-му очку.

Победитель лиги – это команда, которая набрала наибольшее количество очков за весь сезон.

Для каждой из N команд футбольной лиги известно количество побед и ничьих. Вам нужно найти команду-победителя и вывести количество набранных ей очков.

Входные данные. В первой строке входных данных задано - количество команд N ($1 \leq N \leq 50$).

В следующих N строках задано по два целых числа A и B ($0 \leq A, B \leq 100$), количество побед и количество ничьих для i - ой команды.

Выходные данные. В единственной строке выведите очки победителя лиги.

№	input.txt	output.txt
1.	5 10 35 10 10 10 35 10 0 10 7	65
2.	5 1 3 4 1 3 5 0 3 0 1	14

Пояснение для второго теста.

*Команда 1: $3*1 + 1*3 = 6$ очков*

*Команда 2: $3*4 + 1*1 = 13$ очков*

*Команда 3: $3*3 + 1*5 = 14$ очков*

*Команда 4: $3*0 + 1*3 = 3$ очков*

*Команда 5: $3*0 + 1*1 = 1$ очка*

Команда 3 - победитель лиги.

Решения:

Pascal ABC

```
program task_103;  
var n,a,b,s,m,i:integer;  
begin  
  readln(n);  
  m:=0;
```

```

for i:=1 to n do
  begin
    readln(a,b);
    s:=a*3+b;
    if s>m then m:=s;
  end;
write(m);
end.

```

Задача №104. Коробка для яиц

Есть два типа коробок для яиц. В коробке первого типа помещается **6** яиц, в коробке второго типа **8**. Азамат хочет купить ровно **N** яиц. Необходимо найти минимальное число коробок для яиц, которое нужно купить Азамату.

Входные данные. В единственной строке задано одно целое число **N** ($1 \leq N \leq 100$).

Выходные данные. В единственной строке минимальное число коробок для яиц, которое нужно купить Азамату. Если это сделать невозможно, то вывести **-1**.

№	input.txt	output.txt
1.	20	3
2.	24	3
3.	4	-1

Пояснение для первого теста. Азамат должен купить 2 коробки первого типа и 1 коробку второго типа. $2*6+1*8=20$.

Для второго теста. 1) Азамат может купить 4 коробки первого типа и 0 коробок второго типа. $4*6+0*8=24$ (4 коробок); 2) Азамат может купить 0 коробок первого типа и 3 коробок второго типа. $0*6+3*8=24$ (3 коробок)

Для третьего теста. Ответ -1 потому что он не может купить ровно 15 яиц.

Решения:

Pascal ABC

```
program task_104;
var n,i,s:shortint;
begin
  read(n);
  s:=-1;
  for i:=n div 8 downto 0 do
    if (n-i*8) mod 6=0
      then begin
        s:=i+(n-i*8) div 6;break;
      end;
  write(s);
end.
```

Задача №105. Умножения строки

Введём операцию умножения непустой строки **S** на целое число **K** следующим образом:

1. Если **K>0**, то **S·K** является конкатенацией **K** повторов **S**.
2. Если **K<0**, то **S·K** является конкатенацией **K** повторов развёрнутой **S**.

Входные данные. В первой строке задано целое число **K** ($-50 \leq K \leq 50$). В второй строке задана строка, состоящая из латинских букв.

Выходные данные. В единственной строке выведите ответ на задачу.

№	input.txt	output.txt
1.	-2 sony	ynosynos
2.	2 bolot	bolotbolot
3.	4	BooBooBooBoo

	Boo	
4.	-5 Racecar	racecaRracecaRracecaRracecaRracecaR
5.	1 Krsu	krsu
6.	-1 bishkek	kekhsib

Решения:

<i>Pascal ABC</i>

```

program task_105;
var k,i:integer;
    s,s1:string;
begin
  readln(k);
  readln(s);
  if k>0 then for i:=1 to k do write(s)
  else
  begin
    s1:="";
    for i:=1 to length(s) do s1:=s[i]+s1;
    for i:=1 to -k do write(s1);
  end;
end.

```

Задача №106. Диапазоны

Дан набор положительных целых чисел. Числа наборе различны, и они даются в порядке возрастания. Для любых двух натуральных чисел $a \leq b$ определим диапазон $[a, b]$ как последовательность всех целых чисел, лежащих между a и b , включительно. Например, $[3,3]=3$ и $[4,7]=4,5,6,7$.

Вы хотите представить исходный набор, как объединение нескольких диапазонов.

Найдите минимальное количество диапазонов, которое нужно для этой цели.

Входные данные. На первой строке - одно натуральное число **K**, показывающее количество чисел в наборе. Следующая строка содержит **K** чисел в возрастающем порядке. $1 \leq K \leq 50$, $1 \leq \text{Числа, наборе} \leq 50$.

Выходные данные. Одно натуральное число.

№	<i>input.txt</i>	<i>output.txt</i>
1.	10 1 2 3 4 5 6 7 8 9 10	1
2.	6 1 6 10 20 32 49	6
3.	10 2 4 5 6 8 9 10 11 12 15	4
4.	15 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47	14
5.	18 10 11 12 13 14 15 20 21 22 23 25 27 28 29 30 31 32 33	4
6.	1 33	1

Примечание: В первом примере можно представить набор как один диапазон [1,10]. Пятом примере четыре [10,15]-[20,23]-[25]-[27,33].

Решения:

Pascal ABC

```

program task_106;
var k,a,s,i,b:byte;
begin
  readln(k);
  read(a);
  s:=1;
  for i:=2 to k do

```

```

begin
  read(b);
  if b-a>1 then s:=s+1;
  a:=b;
end;
write(s);
end.

```

Задача №107. Многочлен

Были выбраны целые числа $A_5, A_4, A_3, A_2, A_1, A_0$ в диапазоне $-10..10$, составлен многочлен $F(X)=A_5*X^5+A_4*X^4+A_3*X^3+A_2*X^2+A_1*X+A_0$ и вычислены значения $F(1), F(2), F(3), F(4), F(5), F(6)$.

Требуется по этим шести значениям найти $F(7)$. Гарантируется, что задача имеет решение.

Входные данные. Шести строк, на каждой одно целое число – значения функции F . Первой строке значение функции $F(1)$, потом $F(2)$ и так далее.

Выходные данные. Одно целое число – значение $F(7)$.

№	<i>input.txt</i>	<i>output.txt</i>
1.	-5 -2 1 4 7 10	13

Решения:

Pascal ABC

```

program task_107;
const n=6;

```



```

var i,j,k,c:longint;
    p:longint;
    a:array[1..n,1..n] of real;
    b:array[1..n] of real;
    x:array[1..n] of real;
    h,s:real;
begin
  for i:=1 to n do
    begin
      p:=1;
      read(c);b[i]:=c;
      for j:=1 to n do
        begin
          a[i,j]:=p;
          p:=p*i;
        end;
      end;
  for i:=1 to n do
    begin
      for j:=i+1 to n do
        begin
          a[j,i]:=-a[j,i]/a[i,i];
          for k:=i+1 to n do a[j,k]:=a[j,k]+a[j,i]*a[i,k];
          b[j]:=b[j]+a[j,i]*b[i];
        end;
      end;
  x[n]:=b[n]/a[n,n];
  for i:=n-1 downto 1 do
    begin
      h:=b[i];
      for j=i+1 to n do
        h:=h-x[j]*a[i,j];
      x[i]:=h/a[i,i];
    end;
  s:=0;

```

```
for i:=n downto 1 do
s:=s*7+x[i];
writeln(s:0:0);
end.
```

Задача №108. Трамп и заборы

Трамп хочет построить забор на границе с Мексикой. На постройку забора нужны кирпичи. Для этого Трамп должен выбрать самые дешевые и качественные кирпичи, потому что Правительство США не хочет тратить много денег на этот забор.

На тендер подали заявки **N** компаний. У каждой компании есть имя – **S**, цена кирпичей **X** на постройку длиной **1** метр и качества **Y**. Помогите Трампу определить победителя тендера, у которого самый дешевый, а затем самый качественный кирпич.

Входные данные. В первой строке число **N** – количество компаний. В следующих **N** строках ($0 < N \leq 10^6$) идут через пробел – **S_i, X_i, Y_i**, где **S_i** – имя **i**-той компании (длина строки не превышает 100 символов и состоит из латинских больших букв и знака подчеркивания “_”), **X_i** ($0 \leq X_i < 2 \cdot 10^9$) цена кирпичей **i**-той компании на постройку забора длины **1** метр и **Y_i** ($0 \leq Y_i < 2 \cdot 10^9$) качество кирпичей **i**-той компании.

Выходные данные. Выведите имя компании – победителя тендера. Победитель всегда определяется однозначно.

№	input.txt	output.txt
1.	4 BAYABDI 1 2 BAYKARIM_INDESTRIS 3 5 TIMELY_SOFT 2 10 MEXICAN_KIRPICH 1 1	BAYABDI

Решения:

Pascal ABC

```

program task_108;
var s,s1,s2,s3:string;
    a,b,n,i,a1,b1:longint;
    code:integer;
begin
  readln(n);
  readln(s);
  s1:=copy(s,1,pos(' ',s)-1);
  s2:=s1;
  delete(s,1,pos(' ',s));
  s1:=copy(s,1,pos(' ',s)-1);
  val(s1,a1,code);
  delete(s,1,pos(' ',s));;
  val(s,b1,code);
  for i:=2 to n do
  begin
    readln(s);
    s1:=copy(s,1,pos(' ',s)-1);
    s3:=s1;
    delete(s,1,pos(' ',s));
    s1:=copy(s,1,pos(' ',s)-1);
    val(s1,a,code);
    delete(s,1,pos(' ',s));;
    val(s,b,code);
    if (a<a1) or (a=a1) and (b>b1)
      then begin

```

```

        s2:=s3;
        a1:=a;
        b1:=b;
    end;
end;
writeln(s2);
end.

```

Задача №109. Дроби

Найти все правильные несократимые дроби, знаменатели которых не превышают **N**.

Входные данные. В первой строке даны одно число **N** ($1 \leq N \leq 10^3$).

Выходные данные. Список дробей в виде **A/B**, по одной в каждой строке, в порядке возрастания строк.

№	input.txt	output.txt
1.	3	1/2 1/3 2/3

Решения:

Pascal ABC

```

program task_109;
var i,j,n:integer;

function Nod(m,n:integer):integer;
begin
    repeat
    if m>n then m:=m mod n
    else n:=n mod m;
    until (m=0) or (n=0);
    Nod:=m+n;

```

```
end;
```

```
begin  
  read(n);  
  for i:=1 to n do  
    for j:=i+1 to n do  
      if Nod(i,j)=1 then  
        writeln(i,'/',j)  
      end.  
end.
```

Задача №110. Взаимно простые

Напишите программу, которая определяет, являются ли два переданные ей числа взаимно простыми, то есть, не имеющими общих делителей, кроме **1**.

Входные данные. Даны два числа **A, B** от 1 до 10^8 .

Выходные данные. Вывести «**YES**» - если являются взаимно простыми, «**NO**» - если нет.

№	input.txt	output.txt
1.	28 15	YES
2.	28 16	NO

Решения:

Pascal ABC

```
program task_110;  
var a,b:longint;  
  
function Nod(m,n:longint):longint;  
begin  
  repeat  
    if m>n then m:=m mod n  
      else n:=n mod m;  
  until (m=0) or (n=0);  
end;
```

```

Nod:=m+n;
end;

begin
  read(a,b);
  if Nod(a,b)=1 then write('YES') else write('NO')
end.

```

Задача №111. Головоломка

Петя разгадывает головоломку, которая устроена следующим образом. Дана квадратная таблица размера $N \times N$, в каждой клетке которой записана какая-нибудь латинская буква. Кроме того, дан список ключевых слов. Пете нужно, взяв очередное ключевое слово, найти его в таблице. То есть найти в таблице все буквы этого слова, причем, они должны быть расположены так, чтобы клетка, в которой расположена каждая последующая буква слова, была соседней с клеткой, в которой записана предыдущая буква (клетки называются соседними, если они имеют общую сторону — то есть соседствуют по вертикали или по горизонтали). Например, на рисунке показано, как может быть расположено в таблице слово olympiad.

P	O	L	T	E
R	W	Y	M	S
O	A	I	P	T
B	D	A	N	R
L	E	M	E	S

Когда Петя находит слово, он вычеркивает его из таблицы. Использовать уже вычеркнутые буквы в других ключевых словах нельзя. После того, как найдены и вычеркнуты все ключевые слова, в таблице остаются еще несколько букв, из которых Петя должен составить слово, зашифрованное в головоломке.

Помогите Пете в решении этой головоломки, написав программу, которая по данной таблице и списку ключевых слов выпишет, из каких букв Петя должен

сложить слово, то есть какие буквы останутся в таблице после вычеркивания ключевых слов.

Входные данные. в первой строке записаны два числа N ($1 \leq N \leq 10$) и M ($0 \leq M \leq 100$). Следующие N строк по N заглавных латинских букв описывают ребус. Следующие M строк содержат слова. Слова состоят только из заглавных латинских букв, каждое слово не длиннее 100 символов. Гарантируется, что в таблице можно найти и вычеркнуть по описанным выше правилам все ключевые слова.

Выходные данные. выведите в алфавитном порядке оставшиеся в таблице буквы.

№	<i>input.txt</i>	<i>output.txt</i>
1.	5 3 POLTE RWYMS OAIPT BDANR LEMES OLYMPIAD PROBLEM TEST	AENRSW
2.	3 2 ISQ ABC IQW I IS	ABCQQW

Решения:

Pascal ABC

```
program task_111;
var i,j:integer;
    n,m:integer;
```

```

    ch : string [200];
    s : string;
begin
assign (input, 'input-111.txt');
assign (output, 'output-111.txt');
reset (input);
readln (n, m);
setlength (s, 1000);
    s := "";
for i := 1 to n do
begin
readln (ch);
s := s + ch;
end;
for i := 1 to m do
begin
readln (ch);
j := 1;
while (length (ch) >= j)
do
begin
// удаляем очередной символ из s
delete (s, pos (ch [j], s), 1);
inc (j);
end;
end;
close (input);
// сортируем оставшуюся строку в алфавитном порядке
for i := 1 to length (s) - 1 do
for j := 1 to length (s) - i do
if (s [j] > s [j + 1])
then
begin
ch [1] := s [j];
s [j] := s [j + 1];

```



```

s [j + 1] := ch [1];
end;
rewrite (output);
write (s);
close (output);
end.

```

Задача №112. Налог

Требуется уплатить налог **S2** сом и **T2** тыйын «не менее **N** процентов» с суммы **S1** сом и **T1** тыйын. Все числа - целые.

Входные данные. Число **S1** из **1..2018**, число **T1** из **0..99**, число **N** из **1..49**, разделенные одинарными пробелами.

Выходные данные. Числа **S2** и **T2**, разделенные пробелом.

№	<i>input.txt</i>	<i>output.txt</i>
1.	52 39 10	5 24

Решения:

Pascal ABC

```

program task_112;
var s1,t1,n:longint;
begin
  readln(s1,t1,n);
  s1:=(s1*100+t1)*n+90;
  writeln(s1 div 10000,' ',s1 div 100 mod 100)
end.

```

Задача №113. Сумма цифр числа

Пусть дано натуральное число. Требуется найти сумму его цифр.

Входные данные. В первой строке входных данных – число N ($0 \leq N \leq 1000000$).

Выходные данные. Вывести сумму цифр числа.

№	<i>input.txt</i>	<i>output.txt</i>
1.	12345	15

Решения:

Pascal ABC

```
program task_113;
var a:longint;
    x,s:integer;
begin
  read(a);
  s:=0;
  while a>0 do
  begin
    x:=a mod 10;
    a:=a div 10;
    s:=s+x;
  end;
  write(s);
end.
```

Задача №114. Количество нечетных и четных

Дано натуральное число. Требуется посчитать количество четных и нечетных цифр в числе.

Входные данные. Целое число N от 1 до 10^9 .

Выходные данные. Два целых числа, через пробел. Первое количество нечетных цифр в числе. Второе – количество четных цифр.

№	<i>input.txt</i>	<i>output.txt</i>
1.	5683	2 2
2.	38012732	4 4
3.	333	3 0

Решения:

Pascal ABC

```
program task_114;
var n:longint;
    s1,s2:integer;
begin
read(n);
s1:=0; s2:=0;
while n>0 do
begin
if (n mod 10) mod 2=1 then s1:=s1+1;
else s2:=s2+1;
n:=n div 10;
end;
write(s1,' ',s2);
end.
```

Задача №115. DVD-диски

До прихода гостей Фарид подготовил самые интересные DVD –диски. В торопях остальные диски были разбросаны на столе. Чтобы скрыть беспорядок, творящийся на столе, младшая сестра Фариды прикрыла стол полотенцем. Стол прямоугольной формы длиной – **X**

и шириной равной Y . Радиус одного диска – R , общее количество дисков равно N единиц.

Все диски расположены на столе и ни один из них не выходит за край стола. Все диски прикрыты одним полотенцем возможного минимального размера. Стороны полотенца расположены параллельно сторонам стола. Определите площадь непокрытой полотенцем поверхности стола.

Входные данные. В первой строке входного файла записаны четыре целых числа – X, Y, N, R , ($0 < N, Y < 1000; 0 < R < 1000; 0 \leq N \leq 1000$). В следующих N строках расположены координаты центра дисков – x_i, y_i .

Выходные данные. Во выходном файле выведите площадь непокрытой полотенцем поверхности стола, свободной от дисков.

№	<i>input.txt</i>	<i>output.txt</i>
1.	10 10 1 3 1 1 2 8 6 4	36

Решения:

Pascal ABC

Сначала найдем по координатам x_i минимальные и максимальные элементы, потом по координатам y_i минимальные и максимальные элементы. Длину сторон полотенца можно вычислить по формулам

$$(maxx+r)-(minx-r)$$

$$(maxy+r)-(miny-r).$$

Следовательно, если вычесть из общей площади площадь полотенца, то мы приходим к решению поставленной задачи.

```

program task_115;
uses Math;
const kol=1000;
var f,g:text;
    i,a,b,n,r:shortint;
    maxx,maxy,minx,miny:shortint;
    s1,s2:shortint;
    x,y:array[1..kol] of shortint;
begin
assign(f,'dvd-inp.txt');
reset(f);
readln(f,a,b,r,n);
for i:=1 to n do readln(f,x[i],y[i]);
close(f);
maxx:=0; maxy:=0;
minx:=1000; miny:=1000;
for i:=1 to n do
begin
minx:=min(minx,x[i]);
miny:=min(miny,y[i]);
maxx:=max(maxx,x[i]);
maxy:=max(maxy,y[i]);
end;
s1:=((maxx+r)-(minx-r))*((maxy+r)-(miny-r));
s2:=a*b;
assign(g,'dvd-out.txt');
rewrite(g);
writeln(g,s2-s1);
close(g);
end.

```

Задача №116. Проверка уравнений

Молодая учительница начальной школы объяснила детям решение уравнений вида $a \cdot x + b = c$ и решила провести контрольную работу. Она написала несколько наборов значений a , b и c , но вовремя сообразила, что ее ученики еще не знакомы с дробями и отрицательными числами. Выясните, сколько наборов определяют уравнения, имеющие целые неотрицательные решения.

Входные данные. В первой строке одно целое число N ($1 \leq N \leq 100$). Далее N строк по три числа в каждой строке, разделенные пробелами, – наборы значений a , b и c . Все числа неотрицательные целые, не превышающие 10^{18} .

Выходные данные. В единственной строке целое число – количество наборов, которые определяют уравнения, имеющие целые неотрицательные решения.

№	input.txt	output.txt
1.	3 1 2 3 4 5 6 7 8 9	1

Решения:

Pascal ABC

```
program task_116;  
var n,i,s:byte;  
    a,b,c:longint;  
begin  
  readln(n);  
  s:=0;  
  for i:=1 to n do  
  begin
```

```

readln(a,b,c);
if (a<>0) and ((c-b) mod a=0) and ((c-b) div a>=0) then inc(s);
if (a=0) and (b=c) then inc(s);
end;
writeln(s);
end.

```

Задача №117. Остатки

Пусть дано **N** целых чисел и целое число **A**. При делении этих чисел на **A** получается новый набор чисел – остатки от деления каждого числа на **A**. Например, у нас есть числа 1, 3, 4 и число 2. После деления на 2 у нас получаются остатки 1, 1, 0. Требуется написать программу, которая находит количество различных чисел в наборе остатков после деления на **A**.

Входные данные. В первой строке содержится два числа **N** и **A**. Где $1 \leq N \leq 10000$ и $1 < A < 2^{31}$. В следующей строке содержится **N** целых чисел, где каждое число лежит в диапазоне от 1 до 2^{31} .

Выходные данные. Вывести одно целое число – количество различных чисел в наборе остатков после деления на **A**.

№	<i>input.txt</i>	<i>output.txt</i>
1.	10 11 1 2 3 4 5 6 7 8 9 10	10
2.	10 17 17 34 51 68 85 102 119 136 153 170	1

Решения:

Pascal ABC

```

program task_117;
var i,j,n,s:integer;
    a,b:longint;

```

```

    f:boolean;
    ost:array[1..10000] of longint;
begin
    readln(n,a);
    fillchar(ost,sizeof(ost),0);
    for i:=1 to n do
        begin
            read(b);
            ost[i]:=b mod a;
        end;
    s:=1;
    for i:=2 to n do
        begin
            f:=True;
            for j:=1 to i-1 do
                if ost[i]=ost[j] then
                    begin
                        f:=false;
                        break;
                    end;
            if f then inc(s);
        end;
    writeln(s);
end.

```

Задача №118. Два треугольника

Два треугольника заданы координатами вершин на плоскости. Определите, можно ли путем параллельных переносов, поворотов и отражений относительно произвольной прямой составить из них прямоугольник.

Входные данные. В первой строке последовательно три пары чисел – описание первого треугольника. Во второй строке, аналогично, описание второго

треугольника. Все шесть чисел в каждой строке разделены пробелом.

Каждая пара чисел - декартовы координаты вершины треугольника на плоскости, все координаты по модулю не превышают 10^6 . Гарантируется, что оба треугольника невырожденные.

Выходные данные. В единственной строке вывести «YES», если прямоугольник составить можно, и «NO» в противном случае.

№	<i>input.txt</i>	<i>output.txt</i>
1.	0 0 10 0 0 12 10 10 22 10 10 20	YES
2.	0 0 10 0 0 15 10 10 24 0 0 21	NO

Решения:

Pascal ABC

```

program task_118;
var
x11,y11,x12,y12,x13,y13,x21,y21,x22,y22,x23,y23,d11,d12,d13,d21,d
22,d23: int64;

function ras(a1,b1,a2,b2: int64):int64;
begin
  ras:=sqr(a2-a1)+sqr(b2-b1);
end;

function f(a,b,c: int64):boolean;
begin
  f:=(c=a+b);
end;

procedure swap(var x: int64;var y: int64);
  var t: int64;
```

```
begin
  t:=x;
  x:=y;
  y:=t;
end;
```

```
begin
  readln(x11,y11,x12,y12,x13,y13);
  readln(x21,y21,x22,y22,x23,y23);
  d11:=ras(x11,y11,x12,y12);
  d12:=ras(x11,y11,x13,y13);
  d13:=ras(x12,y12,x13,y13);
  if d11>d12 then swap(d11,d12);
  if d12>d13 then swap(d12,d13);
  if d11>d12 then swap(d11,d12);
  d21:=ras(x21,y21,x22,y22);
  d22:=ras(x21,y21,x23,y23);
  d23:=ras(x22,y22,x23,y23);
  if d21>d22 then swap(d21,d22);
  if d22>d23 then swap(d22,d23);
  if d21>d22 then swap(d21,d22);
  if f(d11,d12,d13) and f(d21,d22,d23) and (d11=d21) and (d12=d22)
  and (d13=d23) then
    writeln('YES')
  else
    writeln('NO');
end.
```

Задача №119. Квадратура круга

Молодой художник Андрей писал картину на холсте и захотел выпить кофе. Его подруга, программистка Катя, принесла ему кофе и большую тарелку с печеньем, споткнулась о разбросанные бумаги, уронила тарелку на почти готовую картину и сверху еще и пролила кофе.

Когда кофе вытерли и подняли тарелку, выяснилось, что картина под тарелкой практически не пострадала, а вот все вокруг безнадежно испорчено. Чтобы успокоить подругу, Андрей сказал, что основная идея картины сохранилась, и вырезал круг по диаметру тарелки. Действительно, круглая картина была не так уж плоха. Однако Катя знала, что заказчик хотел картину в прямоугольной рамке, и решила рассчитать размеры рамки так, чтобы заказчику не было видно, что холст круглый, и при этом рамка имела бы минимально необходимую площадь, и картину не пришлось бы резать еще.

При формализации задачи Катя сделала следующие допущения.

1. Прямоугольная рамка представляет собой прямоугольник с прямоугольным отверстием. Площадь рамки - это разница между площадью внешнего прямоугольника и площадью отверстия.
2. Заказчику не видно, что холст круглый, если окружность, являющаяся границей холста, расположена между внешним и внутренним прямоугольниками рамки, возможно, касаясь их.

Входные данные. В единственной строке одно целое положительное число $R \leq 1\ 000\ 000$ - радиус картины.

Выходные данные. В единственной строке одно целое положительное число - минимально возможная площадь рамки при условии, что длины сторон внешнего и внутреннего прямоугольников рамки - целые.

№	<i>input.txt</i>	<i>output.txt</i>
1.	5	51

Решения:**Pascal ABC**

```
program task_119;
var r,a,s:longint;
begin
readln(r);
a:=trunc(sqrt(2)*r);
if a*a+(a+1)*(a+1)<4*r*r then s:=a*(a+1)
else s:=a*a;
writeln(4*r*r-s);
end.
```

Задача №120. Рекурсивная сумма цифр числа

Написать рекурсивную функцию нахождения суммы цифр заданного числа

Входные данные. Числа A от 1 до 10^{19} .

Выходные данные. Строка рекурсивного вызова и на следующей строке – сумма цифр числа A.

№	<i>input.txt</i>	<i>output.txt</i>
1.	100	0+0+1 1

Решения:**Pascal ABC**

```
program task_120;
var n:int64;

function rec_sum(n:int64):integer;
begin
if n<10 then
begin
writeln(n);
```

```

    rec_sum:=n;
end
else
begin
    write(n mod 10,'+');
    rec_sum:=rec_sum(n div 10)+n mod 10;
end;
end;

begin
readln(n);
write(rec_sum(n));
end.

```

Задача №121. Остаток от деления числа

По данным двум натуральным числам $2 < M < 2018$, $2 < N < 2018$ найти остаток от деления числа $M^5(M^5+17)/2$ на число N (В одном из тестов будет $M < 7$, $N < 100$).

Входные данные. Два натуральных числа, разделенных одним пробелом.

Выходные данные. одно натуральное число.

№	<i>input.txt</i>	<i>output.txt</i>
1.	6 7	6

Решения:

Pascal ABC

```

program task_121;
var m,n,k:int64;
begin
readln(m,n);
k:=m*m;
k:=k*k*m;

```

```

if m mod 2=0 then
  write((((k div 2) mod n)*((k+17) mod n)) mod n)
else
  write((k mod n)*(((k+17) div 2) mod n) mod n);
end.

```

Задача №122. GPS

По некоторому натуральному числу M , были взяты три натуральных числа $0 < X, Y, Z \leq M$ и вычислены квадраты O_2, A_2, B_2 соответствующих расстояний от точки $M(X, Y, Z)$ до точек $O(0, 0, 0)$, $A(M, 0, 0)$ и $B(0, M, 0)$. По данным четырем числам M, O_2, A_2, B_2 найти числа X, Y, Z .

Входные данные. В единственной строке входных данных заданы четыре натуральных числа $2 \leq M \leq 2018$, O_2, A_2, B_2 , разделенных единичными пробелами.

Выходные данные. В единственной строке выходных данных выведите три натуральных числа X, Y, Z , разделенных единичными пробелами.

№	input.txt	output.txt
1.	5 42 57 27	1 4 5

Решения:

Pascal ABC

```

program task_122;
var m,o,a,b,x,y: longint;
    f: boolean;
begin
  readln(m,o,a,b);
  f:=False;
  for x:=1 to m do
  begin
    if f then break;

```

```

for y:=1 to m do
begin
if (o-x*x-y*y=a-(x-m)*(x-m)-y*y)
and (o-x*x-y*y=b-(y-m)*(y-m)-x*x)
then
begin
write(x,' ',y,' ',round(sqrt(o-x*x-y*y)));
f:=True;
break;
end;
end;
end;
end.

```

Задача №123. Стройся

Обычно, на физкультуре, в начале урока, школьники строятся в линию по росту – от самого высокого до самого низкого. После летних каникул, учитель сказал построится в линию, как ученики стояли в прошлом учебном году, до каникул. Конечно, за лето школьники подросли и стоять они должны по-другому, с учетом их нового роста. Требуется определить сколько школьников стоит не на своем месте в строю. Ученик стоит не на своем месте, если его рост больше роста предыдущего ученика. Такой ученик должен исключатся из строя.

Входные данные. В первой строке идет число N ($0 < N \leq 10^3$) – количество учеников. После идут N строк содержащие рост учеников ($0 < N \leq 10^3$) в строю, после летних каникул.

Выходные данные. Вывести количество учеников, которых надо исключить из строя, т.к. они стоят не на своем месте по росту.

№	input.txt	output.txt
1.	7 150 146 148 147 145 143 142	2

Пояснение к примеру: сначала из строя исключается ученик с ростом 148, так его рост больше, чем у ученика перед ним с ростом 146. Затем из строя исключается ученик с ростом 147, т.к. его рост тоже больше, чем у ученика с ростом 146.

Решения:

Pascal ABC

```

program task_123;
  var n,h,h1,s,i:integer;
begin
  readln(n);
  s:=0;
  readln(h);
  for i:=1 to n-1 do
  begin
    readln(h1);
    if h1>h then
    begin
      inc(s);
      continue;
    end;
    h:=h1;
  end;
  write(s);
end.

```


Задача №124. Второй раз

Дан массив из N целых чисел. Каждый элемент повторяется два раза за исключением одного. Найдите это число.

Входные данные. В первой строке идет число N ($1 \leq N \leq 100$). В следующей строке идут N чисел в пределах от -100 до 100.

Выходные данные. Вывести число, которое не повторяется.

№	input.txt	output.txt
1.	5 1 2 1 2 3	3
2.	3 -50 18 -50	18

Решения:

Pascal ABC

```
program task_124;
var n,i,j,s:integer;
    a:array[1..100] of integer;
    f:boolean;
begin
  readln(n);
  for i:=1 to n do
    read(a[i]);
  for i:=1 to n do
    begin
      s:=0;
      for j:=1 to n do
        if a[i]=a[j] then inc(s);
      if s=1 then begin write(a[i]);halt;end;
```

end;
end.

Задача №125. Саморазделимое число

Саморазделимое число — это число, которое делится на каждую содержащуюся в нем цифру. Например, 128 является саморазделимым числом, поскольку $128 \% 1 = 0$, $128 \% 2 = 0$ и $128 \% 8 = 0$. Кроме того, саморазделимому числу не разрешается содержать цифру ноль. Даны два числа **L** и **K**, которые являются границами, в пределах которых надо найти список всех возможных саморазделимых чисел, включая границы. Исходные данные таковы, что ответ всегда существует.

Входные данные. В первой строке идут два числа **L** и **R** ($1 \leq L \leq R \leq 10000$).

Выходные данные. Вывести список всех саморазделимых чисел через пробел. После последнего числа должен быть пробел.

№	input.txt	output.txt
1.	1 22	1 2 3 4 5 6 7 8 9 11 12 15 22

Решения:

Pascal ABC

```
program task_125;  
var l,r,i:integer;  
  
function fd(n:integer):boolean;  
var m,n1:integer;  
    f:boolean;  
begin  
    n1:=n;  
    while n>0 do
```

```

begin
  f:=true;
  m:=n mod 10;
  if (m=0) or (n1 mod m<>0) then
    begin
      f:=false;
      break;
    end;
  n:=n div 10;
end;
fd:=f;
end;

```

```

begin
  readln(l,r);
  for i:=1 to r do
    if fd(i) then
      write(i, ' ');
  end.

```

Задача №126. Создание пар

У вас есть коллекция карт. Каждая карта содержит целое число от **0** до **N-1** включительно. Вам предоставляется массив из **N** элементов. Для каждого **i** у вас есть **card[i]** – штук карт с номером **i**. Вы хотите создать пары карт с одинаковым числом. Каждая карта может использоваться не более одного раза. Найдите наибольшее количество пар, которые вы можете создать.

Входные данные. В первой строке идет число **N** ($0 < N \leq 50$) – количество карт. После идет строка с **N** числами в диапазоне от **0** до **N-1** разделенные пробелами.

Выходные данные. Вывести целое число – наибольшее количество пар, которые вы можете создать.

№	input.txt	output.txt
1.	3 2 2 2	3
2.	3 1 1 1	0
3.	1 5	2
4.	12 43 23 10 39 39 22 22 0 3 4 3 2	102
5.	0 0	0

Объяснение примеру 1. У вас две карты с числом 0, две карты с числом 1 и две карты с числом 2. Вы можете использовать эти карты для формирования трех одинаковых пар карт: 0 с 0, 1 с 1 и 2 с 2

Объяснение примеру 2. Здесь у вас есть три карты с числами 0, 1 и 2 соответственно. Поскольку карты все разные, вы не можете создавать пары. Таким образом, правильное возвращаемое значение равно 0.

Объяснение примеру 3. У вас пять карт. Каждая из них содержит число 0. Используя эти карты, вы можете сформировать не более двух одинаковых пар карт. (Обратите внимание, что вы не можете использовать одну и ту же карту в нескольких парах.)

Решения:

Pascal ABC

```

program task_126;
var n,i:byte;
    s:integer;
    card:array[1..50] of byte;
begin
  readln(n);
  for i:=1 to n do

```

```

read(card[i]);
s:=0;
for i:=1 to n do
  s:=s+card[i] div 2;
write(s);
end.

```

Задача №127. Красота

Определение: строка красива, если она имеет два последовательных одинаковых символа. Примерами красивых строк являются «KEEP», «ZZZZZ» и «TTORR», а следующие не красивы: «A», «GH» и «ABCABCВX».

Вам дается строка S, и вы собираетесь удалить ровно один символ из S. Возможно ли, что новая строка будет красивой? Если да, ответ «**Possible**». В противном случае – «**Impossible**».

Входные данные. Дана строка длиной от 1 до 100 символов состоящая из заглавных букв латинского алфавита ('A'...'Z').

Выходные данные. Вывести «**Possible**», если возможно получить красивую строку и «**Impossible**» – если нет. Обратите внимание, что возвращаемое значение чувствительно к регистру.

№	<i>input.txt</i>	<i>output.txt</i>
1.	VIKING	Possible
2.	BCAB	Impossible
3.	XX	Impossible
4.	AABB	Possible
5.	QWERTYY	Possible
6.	ITHINKYOUAREAHUMAN	Impossible
7.	BOB	Possible

Пример 1. Вы можете удалить «К», чтобы получить строку «VIING». Эта строка красивая, потому что она имеет два последовательных «И».

Пример 2. Вы можете получить только одну из следующих строк: «СAB», «BAB», «BCB» и «BCA». Ни одна из них не красивая, поэтому ответ «Impossible».

Пример 3. После удаления одного символа вы получите строку «X», которая не красива. Обратите внимание, что вам нужно удалить ровно один символ.

Пример 4. Вы можете получить либо «ABV», либо «ABW». Обе эти строки красивы.

Пример 5. Есть несколько красивых строк, которые вы можете получить. Некоторые из них «WERTYY» и «QWERTYY».

Решения:

Pascal ABC

```
program task_127;
var s:string;
    f:boolean;
    i:byte;
begin
  readln(s);
  f:=True;
  for i:=1 to length(s)-2 do
    if (length(s)>2) and ((s[i]=s[i+1]) or (s[i]=s[i+2])
    or (s[i+1]=s[i+2])) then
      begin
        write('Possible');
        f:=False;
        break;
      end;
  if f then
    write('Impossible');
end.
```

Задача №128. Последовательность 49

Последовательность **49, 4489, 444889, ...** является необычной. Каждое следующее число в последовательности получается вставкой в середину предыдущего числа **48**. Вторая особенность последовательности – любое число последовательности являются квадратом некоторого целого числа. Необходимо найти эти числа.

Входные данные. В первой строке входа задано число **N** ($1 \leq N \leq 10^{10000}$) – число из последовательности.

Выходные данные. Вывести целое число, квадрат которого равен числу **N**.

№	<i>input.txt</i>	<i>output.txt</i>
1.	49	7
2.	4489	67

Решения:

Pascal ABC

```
program task_128;
var n:string;
    s,i:integer;
begin
  readln(n);
  s:=0;
  for i:=1 to length(n) do
    if n[i]='4' then inc(s);
  for i:=1 to s-1 do
    write('6');
  write('7');
end.
```

Задача №129. Числа Фибоначчи

Последовательность Фибоначчи определяется следующим образом: $F[0]=0$, $F[1]=1$, для каждого $i \geq 2$: $F[i]=F[i-1]+F[i-2]$

Таким образом, последовательность Фибоначчи начинается следующим образом: 0, 1, 1, 2, 3, 5, 8, 13, ... Элементы последовательности Фибоначчи называются числами Фибоначчи.

Вам дается число N . Требуется преобразовать N в число Фибоначчи. Это изменение будет состоять из нуля или более шагов. На каждом шаге вы можете увеличивать или уменьшать число. То есть на каждом шаге вы можете изменить текущее число X либо на $X+1$, либо на $X-1$. Введите наименьшее количество шагов, необходимых для преобразования N в число Фибоначчи.

Входные данные. Дано целое число N ($1 \leq N \leq 10^6$).

Выходные данные. Целое число – количество шагов необходимых для преобразования N в число Фибоначчи.

№	<i>input.txt</i>	<i>output.txt</i>
1.	1	1
2.	13	0
3.	15	2
4.	19	2

Решения:

Pascal ABC

```
program task_129;
uses Math;
var n,a,b,c:longint;
begin
  readln(n);
```



```

a:=0;
b:=1;
while b<=n do
begin
  c:=a+b;
  a:=b;
  b:=c;
end;
write(min(n-a,b-n));
end.

```

Задача №130. Делители

Дано натуральное число **М**. Получить в порядке возрастания **N** первых натуральных чисел не больше, чем **М**, которые не делятся ни на какие простые числа кроме **2, 3, 5**. У числа должны быть все простые делители **2, 3** и **5**.

Например, число 30 является искомым числом, число имеет все простые делители $2*3*5=30$, а число 10 только 2 и 5 и потому не является искомым числом.

Входные данные. Первая строка содержит два целых числа **М** и **N**. $1 \leq M \leq 10^5$, $1 \leq N \leq 1000$.

Выходные данные. Вывести **N** чисел, по одному в строке в порядке возрастания. Если таких чисел меньше, чем **N**, то вывести их все. Если таких чисел нет, то вывести 0.

№	input.txt	output.txt
1.	10 3	0
2.	60 2	30 60

Решения:

Pascal ABC

```
program task_130;
var n,m,i,s:longint;
begin
  readln(n,m);
  s:=30;
  if n<30 then
    write(0)
  else
    for i:=1 to m do
      begin
        if s>n then break;
        writeln(s);
        inc(s,30);
      end;
end.
```

ИСПОЛЬЗОВАННЫЕ ЛИТЕРАТУРА

1. Осмоналиев А.Б., Аркабаев Н.К. Borland Pascal 7.0. Программалоонун негиздери. Окуу китеби: I бөлүк, 2008. – 256 б.
2. Осмоналиев А.Б., Аркабаев Н.К. Borland Pascal 7.0. Программалоонун негиздери. Окуу китеби: II бөлүк, 2010. – 304 б.
3. Аркабаев Н.К., Панков П.С. Информатика боюнча олимпиадалык маселелер жыйнагы. – Ош: 2019. – 144 б.
4. <http://dist-olimpiada.krasnogorka.edusite.ru/>
5. <http://olymp.krsu.edu.kg>
6. <http://acmp.ru/>

**АРКАБАЕВ Н.К.
КОШМУРЗАЕВ И.Б.**

**РЕШЕНИЕ ОЛИМПИАДНЫХ ЗАДАЧ
ПО ИНФОРМАТИКЕ**

Подписано в печать 05.03.2019 г.
Формат: 60x84 1/16
Объем: 11 п.л.
Тираж: 500 экз.

ПЦ «Максимум»
г. Ош, просп. Масалиева 65
Эл.почта: maximum.osh@mail.ru
Тел.: (03222) 2-35-60
0556 023560

