

✓ Задача №4.

На прямой улице находится $n + 1$ домов с номерами от 0 до n , расположенные в порядке возрастания друг за другом. Стало известно, что грабители планируют проникнуть в некоторые дома на этой улице, поэтому полиция выслала сразу несколько патрульных групп на предполагаемое место преступления. Каждый патруль может охранять сразу несколько домов с подряд идущими номерами, и дом может находиться под охраной сразу нескольких групп одновременно. Патрули также могут передвигаться по улице, при этом количество домов, которые они могут держать под наблюдением одновременно остается неизменным. Если грабитель проникнет в дом, то диспетчеру должно быть доложено количество групп, находившихся на месте преступления в момент проникновения.

Входные данные:

- Первая строка содержит числа n и m , разделенные пробелом.
- m – количество строк, каждая из которых обозначает определенное действие и может иметь один из следующих форматов:
 - **P u v**
Высылает на улицу патрульную группу, которая будет держать под наблюдением дома с u -го по v -й включительно. Каждая высланная группа получает свой порядковый номер, начиная с 1.
 - **M I d**
Диспетчер дает команду группе с номером I перейти на d домов. Если $d < 0$, то группа двигается в начало улицы (к нулевому дому), если $d > 0$ – в конец (к дому с номером n)
 - **V x**
Грабитель пытается проникнуть в дом x

Выходные данные:

При каждой попытке ограбления выводится целое число, показывающее количество патрульных групп, охраняющих дом.

Ограничения:

- $0 < n < 10^9$
- $0 < m \leq 250000$
- $0 \leq u \leq v \leq n$

Пример:

Ввод:	Вывод:
7 5	2
P 1 4	1
P 3 5	
V 3	
M 2 1	
V 3	

Требования к оформлению задач по программированию:

- 1) Программы должны быть написаны на одном из языков: C, C++, Pascal
- 2) Полностью оформленная задача должна содержать:
 - программу, выполняющую необходимые операции для всех допустимых данных;
 - операции с файлами входных и выходных данных или понятный пользователю интерфейс ввода исходных данных;
 - комментарии к тексту программы, облегчающие ее понимание.

Невыполнение вышеуказанных требований влечет за собой снижение получаемых за задачи баллов



787

1	2	3	4	Σ
6	8	5	10	29

заполняется жюри!

58

**ПИСЬМЕННАЯ РАБОТА УЧАСТНИКА
ОЛИМПИАДЫ ШКОЛЬНИКОВ СПбГУ
2016–2017**

заключительный этап

Предмет (комплекс предметов) Олимпиады **ИНФОРМАТИКА (8-9 КЛАССЫ)**

Город, в котором проводится Олимпиада ЧЕЛЯБИНСК

Дата 21.10.2017

Вариант 01

✓ **Задача №1.**

На вход подается строка с математическим выражением, необходимо написать программу или алгоритм на языках C, C++, Pascal, вычисляющую результат выражения. Выражение должно содержать однотипные знаки (+, -), но не менее 5 знаков.

Входные данные: Строка с математическим выражением.

Выходные данные: Число-ответ.

Пример:

Ввод: 2+21-35+10-5+11

Вывод: 4

Требования к оформлению задач по программированию:

- 1) Программы должны быть написаны на одном из языков: C, C++, Pascal
- 2) Полностью оформленная задача должна содержать:
 - программу, выполняющую необходимые операции для всех допустимых данных;
 - операции с файлами входных и выходных данных или понятный пользователю интерфейс ввода исходных данных;
 - комментарии к тексту программы, облегчающие ее понимание.

Невыполнение вышеуказанных требований влечет за собой снижение получаемых за задачи баллов

Задача №2.

Необходимо написать на специальном языке набор команд для построения правильного 8-угольника со всеми его диагоналями.

Доступны следующие команды:

- НЦ_x, КЦ – начало и конец цикла, x – количество выполнений цикла;
- НР, КР – начало рисования, конец рисования;
- ПВРТ_ЛВ_{xx}, ПВРТ_ПР_{xx} – поворот влево или вправо на xx градусов на месте;
- ВП_{xx}, НЗ_{xx} – вперед или назад на xx шагов (1 шаг = 1 см);
- ИСХ – команда возвращения робота в исходную позицию.
- ПРГ_x – команда для создания подпрограммы.

* Допускается создание подпрограмм. Тогда конечная программа будет выглядеть как набор подпрограмм с заданным порядком.

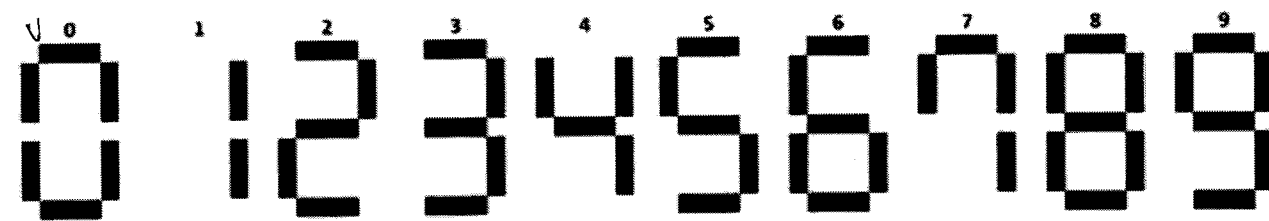
** Программа должна быть компактной и содержать циклические конструкции.

<pre> НР НЦ₈ ВП₁₀₀ ПВРТ_ПР₄₅ КЦ ИСХ НЦ₈ ПВРТ_ПР_{22,5} ВП₁₆₀ ПВРТ_ПР_{22,5} НЗ₁₀₀ КЦ ИСХ НЦ₈ ПВРТ_ПР₄₅ ВП₂₄₀ ПВРТ_ПР₉₀ КЦ ИСХ НЦ₄ ПВРТ_ПР_{67,5} ВП₂₈₀ ПВРТ_ПР_{67,5} НЗ₁₀₀ КЦ ИСХ КР </pre>	<p>НАЧАЛО</p> <p>} РИСУЕМ 8 8-УГОЛЬНИК</p> <p>} РИСУЕМ 1-е 8 ДИАГОНАЛЕЙ МЕЖДУ ВЕРШИНАМИ ЧЕРЕЗ ОДНУ</p> <p>} РИСУЕМ 2-е 8 ДИАГОНАЛЕЙ МЕЖДУ ВЕРШИНАМИ ЧЕРЕЗ ДВЕ</p> <p>} ЗАКАНЧИВАЕМ РИСОВАТЬ ДИАГОНАЛИ.</p>
---	--

(8)

Задача №3.

Цифровое табло, состоящее из семи-сегментных дисплеев, используется для вывода числовых значений. Вы можете активировать выключенные сегменты дисплея, но не можете выключать уже активные. Необходимо написать программу или алгоритм на языках C, C++, Pascal для определения максимального числа, которое может быть выведено на дисплей и не будет больше M.



Входные данные:

Выходные данные:

Целое число и ограничение M. Количество дисплеев равно количеству цифр во введенном числе.

Целое число. Количество цифр в конечном варианте должно быть равно количеству цифр в начальном.

Пример:

Ввод: 25 100

Вывод: 89

Требования к оформлению задач по программированию:

- 3) Программы должны быть написаны на одном из языков: C, C++, Pascal
- 4) Полностью оформленная задача должна содержать:
 - программу, выполняющую необходимые операции для всех допустимых данных;
 - операции с файлами входных и выходных данных или понятный пользователю интерфейс ввода исходных данных;
 - комментарии к тексту программы, облегчающие ее понимание.

Невыполнение вышеуказанных требований влечет за собой снижение получаемых за задачи баллов

// Pascal

[N1]

// ограничение: строка не больше 255 символов

Var s: string;

i, k, p, t, L, Q : integer;

m: array [1..1000] of integer;

Begin

readln(s); // считываем строку

k:=0; // разряд числа

t:=0; // текущее число

p:=0; // отрицательное число

L:=1;

for i:=1 to length(s) do

begin

if s[i]='+' then begin // если плюс, то

m[L]:=t; // число положительно

L:=L+1; // обнуляем и записываем

p:=0;

k:=0;

end;

else if s[i]='-' then begin // если минус, то

m[L]:=#; // число отрицательно

L:=L+1; // обнуляем, записываем

p:=1;

k:=0;

end;

else ~~begin~~ begin k:=k+1; // увелич. цифра

if p=0 then t:=t*k*10+s[i]; // положительное

if p=1 then t:=t*k*10-s[i]; // отрицательное

end;

end;

for i:=1 to L-1 do

Q:=m[i]#Q;

writeln(Q);

end.

// max как мы увелич. разряд
// цифровые

// Вывод.

6

Санкт-Петербургский
Лесотехнический
университет

[N 4]

// Pascal * (ВОЗМОЖНО, int64 РАБОТАЕТ
// ТОЛЬКО В pascal ABC)

var a, b, ~~q~~, m, ~~nom, k, p~~, nom, k, p: ~~int64~~ int64;

~~g, m, p: array [1..250000] of int64;~~ c: char;

g, m, p: array [1..250000] of int64;

begin

q := 0;

readln(m, n);

nom := 0;

for i := 1 to m do

begin

read(c);

if c = 'P' then begin

nom := nom + 1;

read(n[nom]);

read(m[nom]);

end;

if c = 'M' then begin

read(a);

read(b);

n[a] := n[a] + b;

m[a] := m[a] + b;

end;

if c = 'B' then begin read(a); p := 0;

for k := 1 to ~~nom~~ nom do

if (n[k] >= a) and (m[k] >= a) then

p := p + 1;

~~q~~ q := q + 1;

g[q] := p;

end;

end;

for i := 1 to q do writeln(g[i]); // в конце выводит.

// считываем ~~номер~~ ^{улицы} / ~~длина~~ ^{длина}

// считываем буквы

// если "P", то

// добавляем номерок

// от какой ~~длина~~ ^{длина}

// до какой ~~длина~~ ^{длина}

// если "M", то

// считываем n и m

// считываем значения

// проверяем "n"

// проверяем "m"

// если "B", то

// цикл ^{номер} ~~длина~~

// добавляем номер

// до какой ~~длина~~ ^{длина}

// увеличиваем ~~длина~~ ^{длина}

// заносим в массив

N3, programmenue

if $m_i = 2$ then ~~repeat~~ ~~until~~ ~~masko[0]~~ then

repeat

if $m[\text{masko}][0]$ then $ko := ko - 1$

else $m_i = 4$;
if $ko = ko_2$ then $m_i = 1$;
until $(m_i = 4) \text{ or } (m_i = 1)$;

if ydopali summe,
if kompyar go rabotat.

~~if $m_i = 1$ then
~~repeat~~ for $i = ko$ down to 1 do
begin repeat
if $mas[i] > m1[ko_2]$ then
for $k := mas[i] - 1$ down to 0 do
if $m[\text{mas}[i]][k]$ then
begin
 $mas[i] := k$;
 break;
end;
until $mas[i]$~~

$ko_2 := 1$; $a := 69$;
repeat $m1[ko_2] := a \bmod 10$;
 $ko_2 := ko_2 + 1$; $a := a \div 10$;
until $a = 0$;

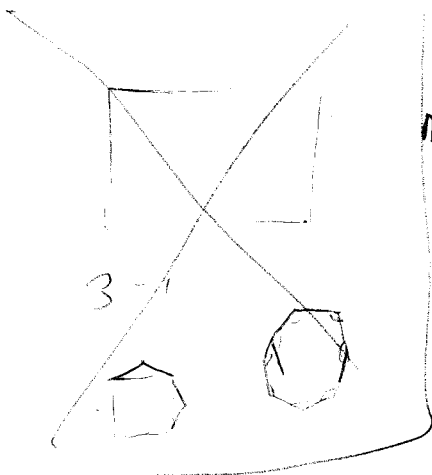
if $m_i \leq 1$ then
for $i = a_1$ down to 0 do
begin

if $a_1 \leq b_1$ then
begin ~~repeat~~ ~~until~~ $ko := 0$;
 $a := i$

repeat
 $mas[ko] := a \bmod 10$;
 $ko := ko + 1$;
 $a := a \div 10$;
until $a = 0$;

for $k := 1$ to 100 do begin
 $mas[k] := 0$;
 $m1[k] := 0$;

Нужно
поставить
массив



```

010x
V
Var m, i, k, a, b, n, ko, ko2; integer;
m, mas: array [0..100] of integer;
m: array [0..9][0..9] of boolean;
f: true; a1: integer;

```

// m - массив преобразований,
// какие числа можно получить

```

begin
m[0][0] := true; m[0][1] := false;
for i := 0 to 9 do
  for k := 0 to 9 do m[i][k] := false;
for i := 0 to 9 do m[i][i] := true;
m[0][8] := true; m[1][0] := true; m[1][3] := true;
m[1][7] := true; m[1][8] := true; m[1][9] := true;
m[2][8] := true; m[3][8] := true; m[3][9] := true;

```

```

for k := 0 to 9 do m[k][8] := true;
m[4][9] := true; m[5][6] := true; m[5][9] := true;
m[7][9] := true;

```

// все true - можно преобразовать

```

read(a); read(b); ko := 1;
a1 := a; b1 := b;
repeat mas[ko] := a mod 10; ko := ko + 1; a := a div 10;
until a = 0;
f := true; ko := ko - 1;

```

// m. крат. дальше

```

while f do ko2 := 1;
begin
  repeat
    m1[ko2] := b mod 10;
    ko2 := ko2 + 1;
    b := b div 10;
  until b = 0;

```

// считываем
// числа

```

mi := 0;
if ko > ko2 then mi := 2;
if ko = ko2 then mi := 1;
if ko < ko2 then mi := 0;

```

// в
// узлом рожу
// в чисел.

// Pascal

// ОГРАНИЧЕНИЯ : СТРОКА не больше 255 СИМВОЛОВ

Var s:string;
i,k,p,t,L : integer
m: array [1..1000] of integer;

Begin

readln(s); k:=0; p:=1; // считывание строки

for i:=1 to length(s) do

begin
if s[i]='+' then begin
p:=1;
k:=0;
end

// если плюс, то
// число положительно
// обнуляем разряд числа [k]

else
if s[i]='-' then begin
m[i]:=t;
p:=0;
k:=0;
end

// иначе
// если минус, то
// записываем число
// отрицательно.

~~else begin k:=k+1; k:=k*10; // цифра
t:=t*10+s[i]; // получаем новое число
k:=k div 10 // получаем новый разряд~~

au groupu m