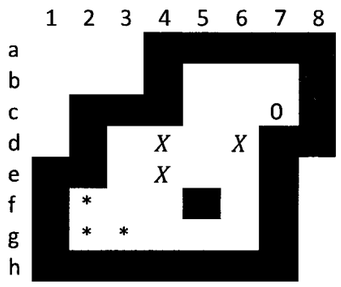
Задача №4.



Вася работает в тренажерном зале. В конце рабочей смены, он осматривает весь зал. Если находит на полу брошенный блин (X), то относит его на свое место (*).

Помогите Васе разработать оптимальный алгоритм для нахождения брошенных блинов в зале и перемещения их на свои места.

** Известно, что с блином Вася может двигаться только вперед.*
*** Вася обязательно должен осмотреть весь зал. За один ход он осматривает одну клетку.*
**** Вася не может брать больше 1го блина.*

Входные данные: (X) – блины, (*) – места хранения блинов, 0 – стартовая позиция Васи.
■ – граница тренажерного зала.

Доступны следующие команды:

НЦ$_x$, КЦ – начало и конец цикла, x – количество выполнений цикла;
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);

** Допускается создание подпрограмм. Тогда конечная программа будет выглядеть как набор подпрограмм с заданным порядком.*
*** Программа должна быть компактной и содержать циклические конструкции.*

ЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

2

5488

1	2	3	4	Σ
6	9	14	∅	29

заполняется жюри!

58

ПИСЬМЕННАЯ РАБОТА УЧАСТНИКА ОЛИМПИАДЫ ШКОЛЬНИКОВ СПбГУ 2016–2017

заключительный этап

Предмет (комплекс предметов) Олимпиады ИНФОРМАТИКА (7 КЛАСС)

Город, в котором проводится Олимпиада _____

Дата _____

****

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:

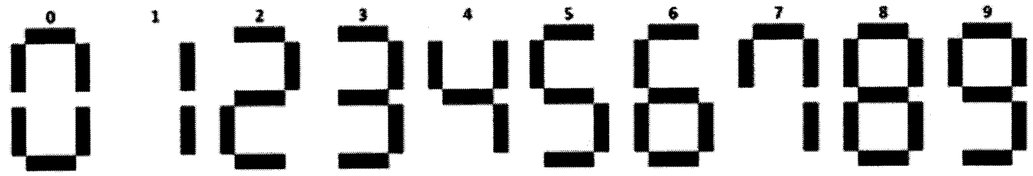
$НЦ_x$, КЦ – начало и конец цикла, x – количество выполнений цикла;
НР, КР – начало рисования, конец рисования;
$ПВРТ\_ЛВ_{xx}$, $ПВРТ\_ПР_{xx}$ – поворот влево или вправо на xx градусов на месте;
$ВП_{xx}$, $НЗ_{xx}$ – вперед или назад на xx шагов (1 шаг = 1 см);
ИСХ – команда возвращения робота в исходную 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.*

```
НЦ8          ← цикл, каждая итерация отрисовывает 1 вершину + все её диагонали
  ПВРТ_ПР67.5
  НЦ5        ← отрисовывает пять диагоналей каждой вершины в цикле
    ВП30
    НЗ30
    ПВРТ_ЛВ22.5
  КЦ
  ВП19       ← отрисовывает сторону (19 подобрано эмпирически, чтобы подходило для каждой диагонали)
  ПВРТ_ПР30
КЦ
```


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:

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

Выходные данные:

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

Задача №1

```
#include <iostream>
using namespace std;
int main()
{
char str[1];
int summ=0;
int current=0;
bool plus=false;
int length;                          //объявление переменных
cout << "Enter the string:  ";
cin >> str;                          //Ввод строки
cout << endl;
Length =(sizeof(str)/sizeof(char))-1; //Вычисляем длину строки
for (int i=0; i<Length ; ++i) {      //начинаем цикл обработки
  if (str[i] == "+") {               //если встретили символ "+"
    if(plus) {                       //... и при этом, предыдущий знак тоже был "+"
      summ += current;               //то прибавляем к общей сумме число, между двумя пол. знаками
      current = 0;                   //и обнуляем его.
    }
    else {
      summ -= current;               //если же встретили "-", то вычитаем число из суммы,
      current =0;                    //также обнуляем его,
      plus = true;                   //,а также перезаписываем переменную последнего знака.
    }
  }
  else if(str[i] == "-"){            //аналогично если встретили "-"...
    if(plus) {
      summ += current;
      current = 0;.
      plus = false;
    }
    else {
      summ -= current;
      current =0;
    }
  }
  else {                             //если встретили не знак, а цифру, то
      current *= 10;                 //... домножаем пред.значение на 10 (сдвиг разряда)
      current += int(str[i]);        //и прибавляем новую цифру.
  }
}
cout << summ << endl;                //Выводим результат
}
```

(6)

Задача №3

```
#include <iostream>
using namespace std;

int increase (int n, int m, bool matrix[10][10]) {
  for (int i=0; i<10; ++i) {
  int current = n;
    if (i=m) return current;
    else if (matrix[n][i]) current = i;
  }
  return current;
}
int main() {
int n,m,ed,dec,ed_m,dec_m;
bool matrix[10][10] = [ [1,0,0,0,0,0,0,0,1,0],
                        [0,1,0,1,1,0,0,1,1,1],
                        [0,0,1,0,0,0,0,0,1,0],
                        [0,0,0,1,0,0,0,0,1,1],
                        [0,0,0,0,1,0,0,0,1,1],
                        [0,0,0,0,0,1,1,0,1,1],
                        [0,0,0,0,0,0,1,0,1,0],
                        [0,0,0,0,0,0,0,1,1,1],
                        [0,0,0,0,0,0,0,0,1,0],
                        [0,0,0,0,0,0,0,0,0,1] ]
cout << "Enter number:   ";      // Ввод данных
cin >> n;
cout << endl;
cout << "Enter M:   ";
cin >> m;
cout << endl;

ed = n%10;                       // Находим единицы и десятки в числе и в М
dec = (n-ed)/10;
ed_m = m%10;
dec_m = (m-ed_m)/10;
if (ed < ed_m + 1) {             // Если единицу в данном числе изначально меньше или равно единице в М
    dec = increase (dec, dec_m+1, matrix);   // И то увеличиваем сначала десятки,
    ed = increase (ed, ed_m+1, matrix);      // а потом единицы
}
else {                           // Если единицу в данном числе изначально больше чем в М
    dec = increase (dec, dec_m, matrix);     // И то десятки увеличиваем не до конца (иначе будет > М)
    ed = increase (ed, 9, matrix);           // А единицы увеличиваем максимально.
}
n = dec*10 + ed                  // Восстанавливаем число и выводим его
cout << n << endl;
}
```

// функция увеличения цифры по имеющуюся границу (m)
// причём, так, что после увеличения цифра м.б. равна m
// но не может быть больше. matrix будет описана
// позже.

// чтобы при увеличении числа, не
// перебирать if'ами кучу вариантов,
// создадим граф (и ао и смежности)
// где будем хранить возможные изменения
// чисел.

(14)