Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

Кафедра ИИТ

**Лабораторная работа №1**

По дисциплине: “Основы алгоритмизации и программирования”

Тема: ”Пользовательские функции”

Вариант №6

**Выполнил:**

Студент 1-го курса

Группы ПО-7

Комиссаров А.Е.

**Проверила:**

Войцехович О.Ю.

Брест, 2020

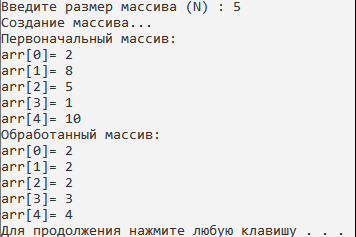
**1) Цель работы:** Изучение параметров функций и реализация выданных задач.

**Задание 1:** Описать процедуру Smooth2(A, N), выполняющую сглаживание вещественного массива A размера N следующим образом: элемент A1 не изменяется, элемент AK (K = 2, …, N) заменяется на полусумму исходных элементов AK–1 и AK. Массив A является входным и выходным параметром. С помощью этой процедуры выполнить пятикратное сглаживание данного массива A размера N, выводя результаты каждого сглаживания.

**Код программы:**

#include <stdio.h>  
#include <iostream>  
#include <iomanip>  
#include <time.h>  
  
using namespace std;  
  
void smooth2(int \*A, int N){  
        int count;  
        int \*A2= new int[N];  
        A2[0]=A[0];  
        for(count=0; count<N; count++){  
            if(count!=0){A2[count]=(A[count]+A[count-1])/2;}  
        }  
        for(count=0; count<N; count++){  
            A[count]=A2[count];  
        }  
}  
  
int main(){  
    setlocale(LC\_ALL, "Russian");  
    int n, x;  
    cout<<"Введите размер массива (N) : ";  
    cin>>n;  
      
    cout<<"Создание массива..."<<endl;  
    int \*arr=new int[n]; for(int x=0; x<n; x++){      
        arr[x]=rand()%10+1;    } //cоздание массива и зполнение его случ. числами.  
      
    cout<<"Первоначальный массив: "<<endl;  
    for(x=0; x<n; x++){      
        cout<<"arr["<<x<<"]= "<<arr[x]<<endl; } //вывод первоначального массива.  
    for(x=0; x<5; x++){      
        smooth2(arr, n); } //обработка 5 раз.  
      
    cout<<"Обработанный массив: "<<endl;  
    for(x=0; x<n; x++){      
        cout<<"arr["<<x<<"]= "<<arr[x]<<endl; } //вывод обработанного массива.  
      
    delete [] arr; //удаление массива из памяти  
      
    system("pause");  
    return 0;  
}

**Результат выполнения:**

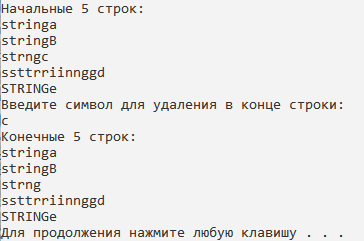
****

**Задание 2:** Описать процедуру TrimRightC(S, C), удаляющую в строке S конечные символы, совпадающие с символом C. Строка S является входным и выходным параметром. Дан символ C и пять строк. Используя процедуру TrimRightC, преобразовать данные строки.

**Код программы:**

#include <iostream>  
#include <stdio.h>  
#include <string>  
using namespace std;  
string TrimRight(string S, char C){  
    if(S[S.size()-1]==C){S.erase(S.size()-1);}  
    return S;  
}  
int main(){  
    setlocale(LC\_ALL, "Russian");  
    string s1="stringa";  
    string s2="stringB";  
    string s3="strngc";  
    string s4="ssttrriinnggd";  
    string s5="STRINGe";  
    char sim;  
    cout<<"Начальные 5 строк: "<<endl;  
    cout<<s1<<endl<<s2<<endl<<s3<<endl<<s4<<endl<<s5<<endl;  
      
    cout<<"Введите символ для удаления в конце строки: "<<endl;  
    cin>>sim;  
  
    s1=TrimRight(s1, sim);  
    s2=TrimRight(s2, sim);  
    s3=TrimRight(s3, sim);  
    s4=TrimRight(s4, sim);  
    s5=TrimRight(s5, sim);  
      
    cout<<"Конечные 5 строк: "<<endl;  
    cout<<s1<<endl<<s2<<endl<<s3<<endl<<s4<<endl<<s5<<endl;  
  
    system("pause");  
}

**Результат выполнения:**

****

**Задание 3:** Описать рекурсивную функцию Combin1(N, K) целого типа, находящую C(N, K) — число сочетаний из N элементов по K — с помощью рекуррентного соотношения:

                               C(N, 0) = C(N, N) = 1,12

              C(N, K) = C(N – 1, K) + C(N – 1, K – 1) при 0 < K < N.

   Параметры функции — целые числа; N > 0, 0 ≤ K ≤ N. Дано число N и пять различных значений K. Вывести числа C(N, K) вместе с количеством рекурсивных вызовов функции Combin1, потребовавшихся для их нахождения.

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

int counter;

int combin1(int n, int k)

{

if ((n == k) || (k == 0)) { counter++; return 1; }

counter += 2;

return combin1(n - 1, k) + combin1(n - 1, k - 1);

}

int main(int argc, char\* argv[])

{

int n = 10, k = 3, c;

counter = 0;

c = combin1(n, k);

printf("C(%d,%d)=%d, counter=%d\n", n, k, c, counter);

system("PAUSE");

return 0;

}

**Результат выполнения:**

****

**Вывод:** изучил параметры функции и реализовал заданные задачи.