Контрольная работа.

Программирование

Оглавление

[Рекомендация по выполнению работы 3](#_Toc378589604)

[Задача 1. Расчет по формуле. 4](#_Toc378589605)

[Задача 2. Ветвление. 6](#_Toc378589606)

[Задача 3. Ветвление. 9](#_Toc378589607)

[Задача 4. Циклы. 11](#_Toc378589608)

[Задача 5. Одномерные массивы и указатели. 13](#_Toc378589609)

[Задача 6. Двумерные массивы 19](#_Toc378589610)

[Задача 7. Строки. 24](#_Toc378589611)

[Задача 8. Функции 28](#_Toc378589612)

[Литература 29](#_Toc378589613)

Рекомендация по выполнению работы

Контрольную работу нужно сдать в печатном виде, обязательно указать фИО, номер варианта работы, номер группы. Перед каждым решенным заданием вставить условия. К каждому решению задачу приложить скриншоты работы программы, не менее двух. Не скупить на комментарии в коде.

Оцениваться будет, не только правильность решения задачи, но и стиль написания программы.

Задача 1. Расчет по формуле.

Напишите программу расчета по двум формулам. Любые параметры задает пользователь, кроме . Для всех переменных возьмите тип данных float или double. Математические функции находятся в заголовочном файле <cmath>. Отсутствующие в библиотеке функции выразите через имеющиеся. В решении укажите код рабочей программы и скриншоты работы программы с разными параметрами.

Пример.

#include "stdafx.h"

#include <iostream>

#include <conio.h>

#include <clocale>

#include <cmath>

using namespace std;

void main()

{

 setlocale(LC\_ALL, "Russian");

 float a, z1, z2;

 //a - параметр

 //z1 - первая функция

 //z2 - вторая функция

 cout<< "Введите параметр a:";

 cin>>a;

 cout << "\n z1="<< sin(a);

 cout << "\n z2="<< a\*a;

 cin>>a;

}

Параметр а =2.

**

Варианты работ.

1.
2. ;

Задача 2. Ветвление.

Решите данную задачу использую оператор if … else. Множественный выбор, осуществить вложенным if. В решение указать рабочий код и скриншоты работы программы.

Пример

Написать программу, которая вычисляет частное двух чисел. Программа должна проверять правильность введенных пользователем данных и, если они неверные ( делитель равен нулю), выводить сообщение об ошибке.

Решение

#include "stdafx.h"

#include <iostream>

#include <conio.h>

#include <clocale>

#include <cmath>

using namespace std;

void main()

{

 setlocale(LC\_ALL, "Russian");

 float a, b;

 //a - параметр

 //b - делитель

 cout<<" Введите делимое : ";

 cin >> a;

 cout<<"\n Введите делитель : ";

 cin>> b;

 if (b ==0)

 cout<<"\nВы ошиблись. Делитель не должен быть равен нулю";

 else

 cout<<"\na/b="<<a/b;

 cin>>b;

}

Результат работы программы:



Варианты работ:

1 . Написать программу вычисления площади кольца. Программа должна проверять правильность исходных данных. Ниже приведен рекомендуемый вид экрана во время выполнения программы (данные, введенные пользователем, выделены полужирным шрифтом).

*Вычисление площади кольца.*

*Введите исходные данные:*

*Радиус кольца (см) – >* ***3,5***

*Радиус отверстия (см) ->* ***7***

Ошибка! Радиус отверстия не может быть больше радиуса кольца.

2. Написать программу, которая переводит время из минут и секунд в секунды. Программа должна проверять правильность введенных пользователем данных и в случае, если данные не верные, выводить соответствующее сообщение.

*Введите минуты :* ***3***

*Введите секунды:* ***10000***

*Ошибка! Количество секунд не может быть больше 60.*

3. Написать программу, который проверяет, является ли введенный с клавиатуры год високосным.

4. Написать программу которая решает квадратное уравнение ( – параметры, их вводит пользователь), если дискриминант меньше нуля, выдать ошибку.

5. Написать программу которая рассчитывает скидку. Если сумма покупка больше 2000, то скидка составляет 10%. Если больше 10000, то скидка составляет 25%. Сумму покупки вводит пользователь, на экран нужно вывести сколько всего к оплате.

6. Написать программу вычисления стоимости покупки с учетом скидки. Скидка в 3% предоставляется, если сумма покупки больше 500 руб, в 5% - если сумма больше 1000 руб.

7. Написать программу, которая запрашивает у пользователя два параметра a и b, и проверяет выполнения условия . Если условие выполнено, то вывести на экран «Попал», иначе «Пропал».

8. Написать программу, которая запрашивает у пользователя два параметра a и b, и проверяет выполнения условия и не . Если условие выполнено, то вывести на экран «Попал», иначе «Пропал».

9. Написать программу поверки знания значения степени самого большого простого числа. У пользователя спрашивают степень, если он вводит не правильно, ему сообщается правильный ответ, иначе поздравляют.( На данный момент 07.02.2013 самое большое простое число у пользователя спрашивают степень двойки).

10. Написать программу проверки знания даты начала второй мировой войны. В случае неправильного ответа пользователя, программа должна выводить правильный ответ. Иначе поздравлять его.

11. Напишите программу проверки знания языка программирования С++. Программа должна вывести вопрос и три варианта ответа. Пользователь должен выбрать правильный ответ и ввести его номер. В итоге программа должна сообщить прав или не прав пользователь. (Вопрос придумать самим).

12. Написать программу, которая сравнивает два введенных с клавиатуры числа. Программа должна указать, какое число больше, или, если числа равны, вывести соответствующее сообщение.

13. Написать программу, которая выводит пример на умножение двух однозначных чисел, запрашивает ответ пользователя, проверяет его и выводит сообщение «Правильно!» или «Вы ошиблись» и правильный результат.

14. Написать программу, которая проверяет, является ли введенное пользователем целое число четным.

15. Написать программу, которая проверяет, делится ли на три введенное с клавиатуры целое число.

16. Написать программу, которая проверяет, делится ли на пять введенное с клавиатуры целое число.

17. Написать программу, которая запрашивает у пользователя номер месяца и затем выводит соответствующее название времени года. Если пользователь ввел недопустимое число, вывести сообщение « ошибка ввода данных».

18. Написать программу, которая запрашивает у пользователя номер дня недели и выводит одно из сообщений: «Рабочий день», «Суббота» или «Воскресение».

19. Написать программу, которая вычисляет квадратный корень из вводимого пользователем числа. Если число отрицательное вывести ошибку. Иначе, значение корня.

20. Написать программу, которая после введенного с клавиатуры числа ( в диапазоне от 1 до 99), обозначающего денежную единицу, дописывает слово «рубль в правильной форме. Например, 12 рублей, 21 рубль и т.д.

Задача 3. Ветвление.

Написать программ ,в которой пользователь вводит параметры и проверяется выполнения условия (Условие приведено ниже, в таблицы, выберите условие с соответствии с номером варианта). Если условие выполняется вывести на экран 0, иначе 1. К решению приложить скриншоты с работой программы, с результатами и с 1 и с 0.

Пример.

Условие: Если

Решение

#include <iostream>

#include <conio.h>

#include <clocale>

#include <cmath>

using namespace std;

void main()

{

 setlocale(LC\_ALL, "Russian");

 float a, b, c;

 cout<<" Введите a: ";

 cin >> a;

 cout<<"\n Введите b: ";

 cin>> b;

 cout<<"\n Введите c: ";

 cin>> c;

 if (a+b>=c && a+c>=b && c+b>=a)

 cout<<0;

 else

 cout<<1;

 cin>>b;

}

Результат работы программы:

**

Вариант:

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |
| 18 |  |
| 19 |  |
| 20 |  |

Задача 4. Циклы.

Вычислить и вывести на экран значения ряда . Где x задает пользователя, также пользователь задает номер элемента, до которого будет считаться ряд (n). Приложить результаты работы для разных x и для разного n.

Пример.

Ряд

Решение:

#include "stdafx.h"

#include <iostream>

#include <conio.h>

#include <clocale>

#include <cmath>

using namespace std;

void main()

{

 setlocale(LC\_ALL, "Russian");

 double x, sum=0, t = -1;

 int i,n;

 cout<<" Введите x: ";

 cin>>x;

 cout<<"\nВведите n: ";

 cin>>n;

 for(i=1;i<=n;i++)

 sum = sum + pow(t,i)\* pow(x,i) / i;

 cout<<"\nСумма ряда равна:"<<sum;

 cin>>x;

}





Варианты:







Задача 5. Одномерные массивы и указатели.

Исходные данные для всех вариантов – n вещественных величин. При написании программ можно использовать как динамические, так и не – динамические массивы. Размерность последних задавать именованной константой.

Пример.

Написать программу, которая для целочисленного массива из 100 элементов определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.

//Написать программу которая для целачисленного массива из 100 жлементов

//определяет, сколько положительных элементво располагается между его максимальным и минимальным элементами.

#include "stdafx.h"

#include <iostream>

#include <conio.h>

#include <clocale>

#include <cmath>

#include <cstdlib>

using namespace std;

void main()

{

 setlocale(LC\_ALL, "Russian");

 //объевляем переменную n, и пhосим пользователя задать ее

 cout<<"Введите количество элментов массива:" ;

 int pmas[n];

 // Заполняем случайными числами массив

 for ( int i =0 ; i<n ; i++)

 {

 double a = rand()\*200/RAND\_MAX -100;

 pmas[i]=a;

 }

 cout<<"\nМассив:\n";

 for (int i =0 ; i<n;i++)

 cout<<i+1<<" "<<pmas[i]<<"\n";

 int N\_MAX=0, N\_MIN=0, kol=0;

 // ищем номер максимального и минимального элемента

 for ( int i =0 ;i<n;i++)

 {

 if (pmas[N\_MAX]<pmas[i])

 N\_MAX=i;

 if(pmas[N\_MIN]>=pmas[i])

 N\_MIN=i;

 }

 // Определяем границу просмотра массива для поиска положительных элементов

 int R=N\_MAX, L=N\_MAX;

 if(N\_MAX>N\_MIN)

 L=N\_MIN;

 else

 R=N\_MIN;

 //ищем количество положительных элементов

 for (int i =L; i<=R;i++)

 if(pmas[i]>0)

 kol++;

 cout<<"\nМежду минимальным и максимальным элементом массива, находится; "<<kol;

 cout<<" положительных элементов,\nпричем номер максимального:"<<N\_MAX+1;

 cout<<"\n а номер минимального:"<<N\_MIN+1;

 cin>>n;

}

Результат работы программы:



Пример 2. Написать программу, которая для вещественного массива из n элементов определяет сумму его элементов, расположенных правее последнего отрицательного элемента.

void main()

{

 setlocale(LC\_ALL, "Russian");

 //объевляем переменную n, и просим пользователя задать ее

 int n;

 cout<<"Введите количество элементов массива:";

 cin>>n;

 double \*pmas=new double[n];

 // Заполняем случайными числами массив

 for ( int i =0 ; i<n ; i++)

 {

 double a = rand()\*200/RAND\_MAX -100;

 pmas[i]=a;

 }

 //Выводим массив

 cout<<"\nМассив:\n";

 for (int i =0 ; i<n;i++)

 cout<<i+1<<" "<<pmas[i]<<"\n";

 int N\_OTR\_POSLED=0, summ=0;

 // ищем номер последнего отрицательного элемента

 for ( int i =0 ;i<n;i++)

 {

 if (pmas[i]<0)

 N\_OTR\_POSLED=i;

 }

 //ищем сумму после последнего отрицательного элемента

 for (int i =N\_OTR\_POSLED+1; i<n;i++)

 summ=summ+pmas[i];

 cout<<"\nСумма элементов массива, находящихся после последнего отрицательного равна "<<summ;

 cout<<"\nпричем номер последнего отрицательного:"<<N\_OTR\_POSLED+1;

 delete [] pmas;

 int i;

 cin>>i;

}

Результат работы программы:



Варианты работ

Вариант 1.

1. Найти сумму отрицательных элементов массива.
2. Найти произведение элементов массива. Расположенных между максимальным и минимальным элементами.
3. Упорядочить элементы массива по возрастанию.

Вариант 2.

1. Найти суму положительных элементов массива.
2. Найти произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.
3. Упорядочить элементы массива по убыванию.

Вариант 3.

1. Найти произведение элементов массива с четными номерами.
2. Найти сумму элементов массива, расположенных между первым и последним нулевыми элементами .
3. Преобразовать массив так, чтобы сначала располагались все положительные элементы, а потом – все отрицательные ( элементы, равные 0, считать положительными).

Вариант 4.

1. Найти сумму элементов массива с нечетными номерами.
2. Найти сумму элементов массивов находящихся между первым и последним отрицательным элементом.
3. Сжать элементы, удалив из массива все элементы больше 10. Освободившиеся в конце массива элементы заполнить нулями.

Вариант 5.

1. Найти максимальный элемент массива .
2. Найти сумму элементов массива, расположенных до последнего положительного элемента массива .
3. Сжать элементы, удалив из массива все элементы меньше 10. Освободившиеся в конце массива элементы заполнить нулями.

Вариант 6.

1. Найти минимальный элемент массива.
2. Найти сумму элементов массива, расположенных между первым и последним положительным элементами.
3. Преобразовать массив так, чтобы сначала располагались все нулевые элементы, ап потом все остальные.

Вариант 7.

1. Найти номер максимального элемента массива.
2. Найти произведение элементов массива, расположенных между первым и вторым нулевыми элементами.
3. Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине – элементы, стоявшие в четных позициях.

Вариант 8.

1. Найти максимальный по модулю элемент массива.
2. Найти сумму элементов массива, расположенных между первым и вторым отрицательными элементами .
3. Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 2, а потом все остальные.

Вариант 9.

1. Найти номер минимального элемента массива.
2. Найти сумму элементов массива, расположенных между первым и вторым положительным элементами .
3. Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

Вариант 10.

1. Найти номер минимального по номеру элемента массива.
2. Найти сумму модулей элементов массива, расположенных после первого элемент, равного нулю.
3. Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине – элементы, стоявшие в нечетных позициях.

Вариант 11.

1. Найти номер минимального по модулю элемента массива.
2. Найти сумму модулей элементов массива, расположенных после первого отрицательного элемента.
3. Сжать массив, удалив из него все элемты, величина которых находится в интервале[a,b]. Освободившиеся в конце массива элементы заполнить нулями.

Вариант 12.

1. Найти номер максимального по модулю элемента массива.
2. Найти сумму элементов массива, расположенных после первого положительного элемента.
3. Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале [a,b], а потом – все остальные.

Вариант 13.

1. Найти количество элементов массива, лежащих в диапазоне от A до B.
2. Найти сумму элементов массива, расположенных после максимального элемента.
3. Упорядочить элементы массива по убыванию модулей элементов.

Вариант 14.

1. Найти количество элементов массива, равных 0.
2. Найти сумму элементов массива, расположенных после минимального элемента.
3. Упорядочить элементы массива по возрастанию модулей элементов.

Вариант 15.

1. Найти количество элементов массива, больших С.
2. Найти произведение элементов массива, расположенных после максимального по модулю элемента.
3. Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом все положительные ( элементы, равные 0, считать положительными).

Вариант 16.

1. Найти количество элементов по модулю меньше 14.
2. Найти сумму модулей элементов массива, расположенных после минимального по модулю элемента.
3. Заменить все отрицательные элементы массива их квадратами и упорядочить элементы массива по возрастанию.

Вариант 17.

1. Найти количество положительных элементов массива.
2. Найти сумму элементов массива, расположенных после последнего элемента, равного нулю.
3. Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых не превышает 1, а потом – все остальные.

Вариант 18.

1. Найти количество элементов массива, меньших С.
2. Найти сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.
3. Преобразовать массив таким образом, чтобы сначала располагались все элементы, отличающиеся от максимального не более чем на 20%, а потом – все остальные.

Вариант 19.

1. Найти произведение отрицательных элементов массива.
2. Найти сумму положительных элементов массива, расположенных до максимального элемента.
3. Изменить порядок следования элементов в массиве на обратный.

Вариант 20.

1. Найти произведение положительных элементов массива.
2. Найти сумму элементов массива, расположенных до первого минимального элемента.
3. Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

Задача 6. Двумерные массивы

Пример 1.

Написать программу, которая для целочисленной матрицы определить среднее арифметическое ее элементов и количество положительных элементов в каждой строке.

#include "stdafx.h"

#include <iostream>

#include <clocale>

#include <cstdlib>

#include <iomanip>

using namespace std;

void main()

{

 setlocale(LC\_ALL, "Russian");

 const int n=10, m=20;

 int matrix[n][m], i, j, sred=0;

 double a=0;

 //Заполняем матрицу

 for(i=0;i<n;i++)

 for(j=0;j<m;j++)

 {

 // создаем случайное число от -100 до 100

 a= rand()\*200/RAND\_MAX - 100;

 matrix[i][j] = a;

 }

 //Выводим матрицу

 cout<<"Исходная матрица:\n";

 for(i=0;i<n;i++)

 {

 for(j=0;j<m;j++)

 cout<<setw(4)<<matrix[i][j]<<" ";

 cout<<"\n";

 }

 //Считаем среднее арифметическое матрицы

 for(i=0;i<n;i++)

 for(j=0;j<m;j++)

 sred = sred + matrix[i][j];

 sred = sred/n/m;

 cout<<"Средние арифметическое: "<<sred;

 //количество положительных элементов в каждой строке

 cout<<"\nКоличество положительных элементов в каждой строке";

 for(i=0;i<n;i++)

 {

 cout<<"\n";

 int kol =0;

 for(j=0;j<m;j++)

 if(matrix[i][j]>0)

 kol++;

 cout<<"строка №"<<setw(2)<<i+1<<" - "<<setw(2)<<kol;

 }

 cin>>i;

}

Скриншот работы программы:



Варианты контрольных задач.

Вариант 1.

Дана целочисленная треугольная матрица. Определить:

1. Количество строк, не содержащих ни одного нулевого элемента;
2. Максимальное из чисел, встречающихся в заданной матрице более одного раза.

Вариант 2.

Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента.

Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, располагать их в соответствии с ростом характеристик.

Вариант 3.

Дана целочисленная квадратная матрица. Определить:

1. Количество столбцов, содержащих хотя бы один нулевой элемент.
2. Номер строки, в которой находится самая длинная серия одинаковых элементов.

Вариант 4.

Дана целочисленная квадратная матрица. Определить:

1. Сумму элементов в тех столбцах, которые не содержат отрицательных элементов;
2. Максимум среди сумм элементов диагональей, параллельных главной диагонали матрицы.

Вариант 5.

Дана целочисленная квадратная матрица. Определить:

1. Сумму элементов в тех столбцах, которые не содержат отрицательных элементов;
2. Максимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы.

Вариант 6.

Дана целочисленная прямоугольная матрица. Определить:

1. Сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент;
2. Номера строк и столбцов всех Седловых точек матрицы.

Матрица А имеет Седловую точку , если является минимальным элементом в i-й строке и максимальном в j столбце.

Вариант 7.

Для заданной матрицы размером 8 \* 8 найти такие k, что k-ая строка матрицы совпадает с k-ым столбцом. Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

Вариант 8.

Характеристикой столбца целочисленной матрицы назовем сумму модулей его отрицательных нечетных элементов. Переставляя столбцы заданной матрицы, располагать и в соответствии сростом характеристик.

Найти сумму элементов в тех столбцах, которые содержат хотя бы один отрицательный элемент.

Вариант 9.

Соседями Элемента в матрице назовем элементы . Операция сглаживания матрицы дает новую матрицу того же размера, каждый элемент которой получается как среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы. Построить результат сглаживания заданной вещественной матрицы размером (10\*10). В сглаженной матрице найти сумму модулей элементов, расположенных ниже главной диагонали.

Вариант 10.

Соседями Элемента в матрице назовем элементы .

Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Подсчитать количество локальных минимумов заданной матрицы размером (10\*10).

Найти сумму модулей элементов, расположенных выше главной диагонали.

Вариант 11.

Коэффициенты систему линейных уравнений заданы в виде прямоугольной матрицы. С помощью допустимых преобразований привести систему к треугольному виду.Найти количество строк, среднее арифметическое элементов которых меньше заданной величины.

Вариант 12.

Уплотнить заданную матрицу, удаляя из нее строки и столбцы, заполненные нулями. Найти номер первой из строк, содержащих хотя бы один положительный элемент.

Вариант 13.

Осуществить циклический сдвиг элементов прямоугольной матрицы на n элементов вправо или вниз(в зависимости от введенного режима). N может быть больше количества элементов в строке или столбце.

Вариант 14.

Осуществить циклический сдвиг элементов квадратной матрицы размерности M\*N вправо на k элементов таким образом: элементы 1-й строки сдвигаются в последний столбец сверху вниз, из него – в последнюю строку справа налево, из нее – в первый столбец снизу вверх, из него – в первую строку, для остальных элементов – анлогично.

Вариант 15.

Дана целочисленная прямоугольная матрица. Определить номер первого столбца, содержащий хотя бы один нулевой элемент. Характеристикой строки целочисленной матрицы назовем сумму ее отрицательных четных элементов. Переставляя строки заданной матрицы, располагать их в соответствии с убыванием характеристик.

Вариант 16.

Упорядочить строки целочисленной прямоугольной матрицы по возрастанию количества одинаковых элементов в каждой строке. Найти номер первого из столбцов, не содержащих ни одного отрицательного элемента.

Вариант 17.

Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине – в позиции (2, 2), следующий по величине в (3,3) и т.д. заполнив таким образом всю главную диагональ.

Найти номер первой из строк, не содержащих ни одного положительного элемента.

Вариант 18.

Дана целочисленная прямоугольная матрица. Определить:

1. Количество строк, содержащих хотя бы один нулевой элемент.
2. Номер столбца, в котором находится самая длинная серия одинаковых элементов.

Вариант 19.

Дана целочисленная квадратная матрица. Определить:

1. Сумму элементов в тех строках, которые не содержат отрицательных элеметов.
2. Минимум среди сумм элементов диагоналей, параллельных главной диагонали.

Вариант 20.

Дана целочисленная прямоугольная матрица. Определить:

1. Количество отрицательных элементов в тех строках, которые содержат хотя бы один нулевой элемент.
2. Номера строк и столбцов всех Седловых точек матрицы.

Матрица А имеет Седловую точку , если является минимальным элементом в i-й строке и максимальном в j столбце.

Задача 7. Строки.

Рекомендации по выполнению задачи.

1. Не использовать функции находящиеся в заголовочном файле <string.h> (<cstring>).
2. После каждого преобразовании строки выводите результат на экран.
3. К рабочему коду программы, приложите скриншоты.

Пример 1.

Написать функцию, которая возвращает преобразованную к верхнему регистру строку

#include "stdafx.h"

#include <iostream>

#include <clocale>

#include <cstdlib>

using namespace std;

void main()

{

 setlocale(LC\_ALL, "Russian");

 const int n=1000;

 int i, dlina;

 char str[n], sim;

 cout<<"Введите строку: ";

 cin>> str;

 for (dlina=0;dlina<=n && str[dlina]!= '\0';dlina++);

 cout <<"\nДлина строки: "<<dlina;

 for (i=0;i<dlina;i++)

 if(str[i]>=97 && str[i]<=122)

 str[i]=str[i]-32;

 cout<<"\n"<<str;

 cin >>i;

}

 Результат работы программы:



Пример 2.

Посчитать сколько слов входит в строку.(Слово разделяется символом “\_”).

Варианты работ.

Вариант 1.

Написать программу, которая:

1. Запрашивает пользователя ввести строку .
2. Выводит количество символов строки.
3. Подсчитывает сколько в троке встречается символ "V”.

Вариант 2.

Написать программу, которая:

1. Запрашивает пользователя ввести строку .
2. Выводит количество символов строки.
3. Переворачивает строку.

Вариант 3.

Написать программу, которая:

1. Запрашивает пользователя ввести строку .
2. Выводит количество символов строки.
3. Выводи слова в строке в обратном порядке (слово разделяется символом «\_»).

Вариант 4.

Написать программу, которая:

1. Запрашивает пользователя ввести строку .
2. Выводит количество символов строки.
3. Выводит все слова начинающиеся с заглавных букв.

Вариант 5.

Написать программу, которая:

1. Запрашивает пользователя ввести строку .
2. Выводит количество символов строки.
3. Поменять местами каждые два соседних слова.

Вариант 6.

Написать программу, которая:

1. Запрашивает пользователя ввести строку.
2. Выводит строку в обратном порядке.
3. Проверяет является ли введенная с клавиатуры строка двухзначным числом.

Вариант 7.

Написать программу, которая:

1. Запрашивает пользователя ввести две строки.
2. Объединяет две строки в одну.
3. В объединённой строки подсчитывает количество слов. (слово разделяется символом «\_»).

Вариант 8.

Написать программу, которая:

1. Запрашивает у пользователя ввести строку.
2. Подсчитывает количество слов в строке. (слово разделяется символом «\_»).
3. Находит все слова которые меньше 4 букв.

Вариант 9.

Написать программу, которая:

1. Запрашивает у пользователя ввести строку.
2. Подсчитывает количество слов в строке. (слово разделяется символом «\_»).
3. Удаляет и строки все буквы “J”.

Вариант 10.

Написать программу, которая:

1. Запрашивает у пользователя ввести строку.
2. Удаляет из строки каждое втрое слово. (слово разделяется символом «\_»).

Вариант 11.

Написать программу, которая:

1. Запрашивает у пользователя ввести строку.
2. Выводи слова в алфавитном порядке. (слово разделяется символом «\_»).

Вариант 12.

Написать программу, которая:

1. Запрашивает у пользователя ввести строку.
2. Сортирует строку в алфавитном порядке.

Вариант 13.

Написать программу, которая:

1. Запрашивает у пользователя ввести строку.
2. Проверят содержит ли строка, введенная с клавиатуры подстрака.

Вариант 14.

Написать программу, которая:

1. Запрашивает у пользователя ввести строку.
2. Меняет в строке все заглавные буквы на строчные, а строчные на заглавные.

Вариант 15.

Написать программу, которая:

1. Запрашивает у пользователя ввести строку.
2. Меняет строку таким образом, чтобы четные буквы были Заглавные, а не четные строчные.

Вариант 16.

Написать программу, которая:

1. Запрашивает у пользователя ввести две строки.
2. Создает новую строку из двух введенных, следующим образом. Четные символы берутся из первой строки, нечетные из второй.

Вариант 17.

Написать программу, которая:

1. Запрашивает у пользователя ввести строку.
2. Переворачивает строку.
3. Удаляет из строки все заглавные буквы.

Вариант 18.

Написать программу, которая:

1. Запрашивает у пользователя ввести строку.
2. Вставить после каждой заглавной буквы символ «\_».

Вариант 19.

Написать программу, которая:

1. Запрашивает у пользователя ввести строку.
2. Меняет местами, каждые две буквы (Например слово «Добрый», поменяется на «оДырй»).

Вариант 20.

 Написать программу, которая меняет введенную с клавиатуры строку следующим образом. После каждой гласной буквы вставить букву “s” и эту гласную. Например была строка “cat”, после преобразований получаем “casat”.

Задача 8. Функции

Задания.

Функции и массивы

Решить задачу 5 (Одномерные массивы) и оформить каждый пункт задания в виде функции. Все необходимые данные для функции должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Литература

1. Т.А. Павловская, С/С++ Программирование на языке ввысокого уровня. – СПб.: Питер, 2013. – 461с.

2. Т.А. Павловская, Ю.А. Щупак С/С++. Структурное и объектно-ориентированное программирование: Практикум. – СПб.: Питер 2011 – 352 с.

3. ["Thinking in C++"](http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html) - это сборник из двух томов, Автор Bruce Eckel.

4. "Язык программирования C++. Вводный курс", Стенли Липпман, Жози Лажойе

5. "Основы программирования на С++. Том 1", Стенли Липпман