**Лабораторная работа № 16. Функции пользователя**

В любой программе С/C++ должна быть функция с именем **main** (главная функция или основная программа) с которой начинается решение задачи. Могут также присутствовать *функции пользователя* (подпрограммы). Функция пользователя может быть записана как до основной программы, так и после. В последнем случае вызов функции следует предварять объявлением (прототипом) этой функции.

При вызове функции значения фактических параметров из вызывающей программы пересылаются в формальные параметры, записанные в определении функции пользователя.

|  |  |  |
| --- | --- | --- |
| **Задание** | **Краткие теоретические сведения** | |
| 1. Изучить способы передачи значений *переменных* в функцию, выполнив программы, записанные в правой части.  Записать условие задачи.  Выводит квадраты чисел от 1 до 10. | |  |  |  | | --- | --- | --- | | Передача параметра  по значению | Передача параметра  по указателю | Передача параметра  по ссылке | | **int square(int);**  **void main()**  **{ for(int x=1; x<=10; x++)**  **cout<<square(x)<<' ';**  **}**  **int square(int y)**  **{ return y \* y; }** | **int square(int \*);**  **void main()**  **{ for(int x=1; x<=10; x++)**  **cout<<square(&x)<<' ';**  **}**  **int square(int\* y)**  **{ return \*y \* \*y; }** | **int square(int &);**  **void main()**  **{ for(int x=1; x<=10; x++)**  **cout<<square(x)<<' ';**  **}**  **int square(int& y)**  **{ return y \* y; }** | | *Значение* фактического параметра **x** копируется в локальную переменную **y**, являющую формальным параметром. | *Адрес* фактического параметра **x** передается в указатель **y**. Для получения значения в функции пользователя требуется операция разыменования. | Параметр **y** выступает *псевдонимом* фактического параметра **x**, то есть передаётся сам объект. Нет необходимости в разыменовании указателя. | | |
| 2. Изучить использование *указателей* и *ссылок* как *формальных параметров* функции при работе с *одномерным* *массивом данных*, опробовав работу программ в правой части.  Написать условие задачи.  Ищет сумму всех элементов массива | Если в качестве передаваемого аргумента используется *массив* данных, то возможна передача только по *указателю* или по *ссылке*.  **Имя одномерного массива без индекса** является **адресом элемента** с **нулевым** индексом. При передаче в функцию имя преобразуется в *указатель*, и копия указателя на начало массива передается в функцию. Можно использовать несколько вариантов передачи массива в функцию пользователя.   1. В примере **mas** и **x** − имена массивов, то есть являются указателями.   **int sum(int x[]) // определение функции**  **{ int res = 0;**  **for(int i = 0; i < 5; i++)**  **res += x[i];**  **return res;**  **}**  **int main()**  **{ int mas[5], i;**  **for(i = 0; i < 5; i++)**  **{ mas[i] = i \* i;**  **cout<< mas[i] <<' ';**  **}**  **cout <<endl<< "result=" << sum(mas) << endl;**  **}**  2. В другом варианте определения функции *указатель* используется явно:  **int sum(int \*x)**  Текст функции пользователя и ее вызов прежние.  3. При передаче массива по *ссылке* можно изменять значение указателя на массив.  **int sum(int &x)**  **{ int \*p = &x, res = 0;**  **for(int i = 0; i < 5; i++)**  **res += p[i];**  **return res;**  **}**  Вызов функции: **cout<<endl <<"result="<<sum(\*mas)<<endl;** | |
| 3. В программе, записанной справа, используется меню, разработанное с помощью оператора **switch**, которое позволяет делать выбор между двумя функциями.  Написать условие задачи и комментарии к программе.  Вычисляет среднее арифметическое всех элементов массива | | **#include <iostream>**  **using namespace std;**  **int massiv(int \*M, int ); //прототип**  **void matrix(); //прототип**  **int main(void)**  **{**  **setlocale(LC\_CTYPE, "Russian");**  **int choice;**  **do**  **{**  **cout << "Выберите вариант работы" << endl;**  **cout << "1 - с одномерным массивом" << endl;**  **cout << "2 - с матрицей" << endl;**  **cout << "3 - выход" << endl;**  **cin >> choice;**  **switch (choice)**  **{**  **case 1: int \*M, size, i, av;**  **cout << "Введите размер массива ";**  **cin >> size;**  **M = new int[size];**  **for (i = 0; i < size; i++)**  **{ cout << "Введите " << i + 1 << " число ";**  **cin >> \*(M + i);**  **}**  **av = massiv(M, size);**  **cout << "Результат=" << av << endl;**  **delete[] M;**  **break;**  **case 2: matrix();**  **break;**  **case 3: break;**  **}**  **} while (c != 3);**  **}**  **int massiv(int \*M, int size)**  **{ int sum = 0, avar;**  **for (int i = 0; i < size; i++)**  **sum += \*(M + i);**  **avar = sum / size;**  **return avar;**  **}**  **void matrix()**  **{ int \*\*A, row, col, i, j, m;**  **cout << "Введите число строк матрицы ";**  **cin >> row;**  **cout << "Введите чиcло столбцов ";**  **cin >> col;**  **A = new int\*[row];**  **for (i = 0; i < row; i++)**  **A[i] = new int[col];**  **for (i = 0; i < row; i++)**  **for (j = 0; j < col; j++)**  **{ cout << "Введите A[" << i << "],[" << j << "]= ";**  **cin >> \*(\*(A + i) + j);**  **}**  **m = A[0][0];**  **for (i = 0; i < row; i++)**  **for (j = 0; j < col; j++)**  **if (\*(\*(A + i) + j) > m)**  **m = \*(\*(A + i) + j);**  **cout << "Результат=" << m << endl;**  **for ( i = 0; i < row; i++)**  **delete A[i];**  **delete[] A;**  **}** |

4. В соответствии со своим вариантом написать программы с использованием ***динамических*** массивов и ***функций* *пользователя*** для условий задач из таблицы.

Объединить написанные программы, разработав интерфейс с помощью оператора **switch**. Начальные размерности и значения элементов массивов ввести с клавиатуры в главной функции.

Массивы передать из главной функции в функции пользователя как ***параметры***.

| **№ варианта** | **Условия задач** |
| --- | --- |
| **2** | 1. Ввести вещественные числа в массив **А**[**n**, **m**] (**n**, **m** > 0) и ко всем числам каждой строки прибавить единицу. Если первый элемент строки нулевой, ничего не делать.  **2.** Дана строка. Все русские буквы привести к верхнему регистру, латинские − заменить символом «**?**». Вывести результат на экран. |
| **4** | 1. Ввести целочисленный массив **A**[**n**, **n**] и вычесть из каждой строки предыдущую. Из первой строки вычесть последнюю строку. Получившийся массив вывести.  2. Дана строка. Слова в предложении разделены одним или несколькими пробелами. Слова могут состоять только из цифр или букв. Найти сумму чисел, входящих в строку. |
| **11** | 1. Дан массив размерностью **N** x **M**. Найти максимальный элемент и его позицию. Если максимальных элементов несколько, то вывести их все. Найти сумму элементов ниже главной диагонали.  2. Преобразовать строку: после каждой буквы **c** добавить символ **\*.** |
| **13** | 1. В матрице **A**[**k**, **n**], **k** <= 12, **n** <= 8 поменять местами строку, содержащую элемент **а** со строкой, содержащей элемент **b**.  2. Дано предложение. Составить программу, которая выводит все слова, оканчивающиеся на -**ая**. |