**Лабораторная работа №2**

**Создание собственных функций для массивов, находящихся в постоянной памяти**

**1. Цель работы**

Приобретение навыков в создании собственных функций, позволяющих производить действия с одномерными, двухмерными массивами и строками (символьными массивами).

**2. Постановка задачи**

Разработать функцию, указанную в варианте и вызвать ее в программе.

**3. Методические указания**

Создать и проинициализировать исходный массив/вектор в постоянной памяти в функции main().

Создать функцию по варианту:

- объявить функцию (создать прототип): тип функции, имя функции, список передаваемых типов;

- описать функцию: тип функции, имя функции, список передаваемых формальных параметров с указанием типов, тело функции;

- вывод на экран данных в функции **не** делать. Для векторов и матриц должна быть отдельная функция печати.

В main() вызвать созданную функцию и передать ей фактические значения в соответствии с заголовком функции.

**4. Варианты заданий**

|  |  |  |
| --- | --- | --- |
| №  вар-та | Тип массива  (входные параметры для функции) | Функция должна |
|  | Вектор беззнакового целого типа | Менять местами элементы, находящиеся на четных позициях, с элементами, находящимися на нечетных позициях. |
|  | Вектор знакового целого типа | Упорядочивать элементы массива по возрастанию. |
|  | Вектор символьного типа, символ | Заменить все гласные буквы на символ &. |
|  | Вектор вещественного типа | Найти минимальный по модулю элемент. |
|  | Матрица целого типа | Циклически переставить столбцы. |
|  | Вектор знакового целого типа | Определить сколько раз элементы массива меняют знак. |
|  | Вектор символьного типа | Определить количество слов, начинающихся с гласной буквы. |
|  | Векторы вещественного типа | Получить новый вектор как разность двух векторов. |
|  | Матрица вещественного типа, число вещественного типа | Заменить данные верхнего треугольника числом. |
|  | Матрица целого типа | Вычислить среднее значение каждой строки. |
|  | Вектор знакового целого типа | Поменять местами элементы массива: первый с последним, второй с предпоследним и т.д. |
|  | Вектор символьного типа | Определить количество союзов (а, и) в тексте. |
|  | Матрицы целого типа | Получить новую матрицу как сумму двух матриц. |
|  | Вектор, матрица целого типа | Заполнить матрицу единицами, а главную диагональ – значениями вектора. |
|  | Матрица вещественного типа | Транспонировать на новом месте. |
|  | Матрица и вектор целого типа | Заполнить вектор элементами побочной диагонали матрицы. |
|  | Матрица целого типа | Найти минимальный элемент и его индексы. |
|  | Матрица целого типа | Найти максимальный элемент и его индексы. |
|  | Матрица целого типа | Поменять местами строки и столбцы так, чтобы наибольшим стал элемент с индексами [0][0]. |
|  | Вектор целого типа | Определить количество нулевых элементов. |
|  | Матрица целого типа | Поменять местами строки и столбцы так, чтобы наименьшим стал элемент с индексами [0][0]. |
|  | Вектор целого типа, число целого типа | Все нулевые элементы заменить на число |
|  | Вектор целого типа | Найти среднее значение вектора. |
|  | Матрица вещественного типа, число вещественного типа | Заменить значения нижнего треугольника числом. |
|  | Матрицы вещественного типа | Циклически переставить строки матрицы. |

Пример программы печати вектора и матрицы Приведен ниже.

**5. Содержание отчета**

1. Оформленный титульный лист с названием работы.
2. Постановка задачи.
3. Программа решения задания.
4. Результаты работы программы – копия консольного окна с результатами.

**6. Вопросы**

1. Каким способом можно передать переменную в функцию?

2. Как можно получить несколько переменных разного типа из функции?

3. Что такое прототип функции и для чего он нужен?

**Пример программы печати вектора и матрицы**. Оба массива находятся в *постоянной (статической)* памяти. Массивы всегда передаются в функцию **по адресу**! (по копии адреса). В примере используется объявление и описание функций.

#include <iostream>

using namespace std;

#define M 4

//прототип функции печати вектора

void PechatVectora(int [], int);

//прототип функции печати матрицы

void PechatMatrici(int [][M], int, int);

void main()

{

setlocale(LC\_ALL, "");

//1/ создание вектора в постоянной памяти

const int N = 6;

int vec[N];

for (int i = 0; i < N; i++) vec[i] = i \* 10 + i + 1;

//вызов функции печати с передачей параметров

PechatVectora(vec, N);

//2. создание матрицы в постоянной памяти

int mat[N][M];

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

for (int j = 0; j < M; j++) mat[i][j] = i \* 2 + j\*3 + 1;

//вызов функции печати с передачей параметров

PechatMatrici(mat, N, M);

}

//=================================================================

//функция печати вектора

void PechatVectora(int v[], int NN)

{

printf\_s("Вектор\n");

for (int i = 0; i < NN; i++) printf\_s("%5d",v[i]);

printf\_s("\n");

}

//функция печати матрицы

void PechatMatrici(int m[][M], int I, int J)

{

printf\_s("\nМатрица\n");

for (int i = 0; i < I; i++) {

for (int j = 0; j < J; j++) printf\_s("%5d", m[i][j]);

printf\_s("\n");

}

}

результат работы программы:

/\*

Вектор

1 12 23 34 45 56

Матрица

1 4 7 10

3 6 9 12

5 8 11 14

7 10 13 16

9 12 15 18

11 14 17 20

Для продолжения нажмите любую клавишу . . .

\*/