

Тема : знакомство с С (Си)

МАССИВЫ и ФУНКЦИИ – упражнения

Содержание: [массивы](#) [упражнение 1](#) [упражнение 2](#)

- **Упражнение 1**

Написать программу, которая работает следующим образом:

1. Создается матрица с заданными измерениями (двумерный массив)
2. Матрица заполняется случайными числами
3. Созданная матрица выводится на экран

Использовать следующие функции:

- a. Функция заполнения матрицы
- b. Функция генерирования случайных чисел
- c. Функция вывода матрицы на экран

В функции main могут быть:

- Декларирование необходимых переменных
- Вызовы функций

Возможное решение (без использования функций):

```
/*
Имя файла:   matriks.c
Автор:
Создано:
Обновлено:

Описание:   Создается матрица с заданными размерами
             Матрица заполняется случайными числами
             Содержимое матрицы выводится на экран
*/

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define RIDU 5
#define VEERGE 5
#define MAXVAARTUS 20

int main (void)
{
//декларирование переменных цикла: целые числа
    int i,j;

//декларирование матрицы, для которой
// RIDU - число рядов(строк), VEERGE - число столбцов
    int matriks[RIDU][VEERGE];

//инициализация генератора случайных чисел
```

```

srand(time(0)) ;

//цикл, проходящий все ряды матрицы
for(i=0; i<RIDU; i++)
{
    //цикл, проходящий для данного ряда i
    //все элементы матрицы, находящиеся в столбцах j
    for(j=0; j<VEERGE; j++)
    {
        //заполнение элемента матрицы
        //случайным значением в промежутке 1 ... maxVaartus
        maatriks[i][j]=rand()%MAXVAARTUS + 1;
    }
}

//присваивание переменным цикла первоначального значения
i=j=0;
//цикл по всем рядам матрицы
while (i<RIDU)
{
    //цикл по всем столбцам матрицы
    while (j<VEERGE)
    {
        //выводим элемент матрицы на экран
        printf("%4d", maatriks[i][j]);

        //увеличиваем счетчик столбцов j на 1
        j++;
    }

    //увеличиваем счетчик рядов i на 1
    i++;

    //для нового ряда необходимо обнулить счетчик столбцов j
    j=0;

    // для вывод нового ряда с новой строки используем:
    printf("\n");
}

//при необходимости добавляем следующие строки
//system ("PAUSE");
// или
//getchar(); getchar();

return 0;
}

```

Возможное решение (с использованием функций):

```

/*
Имя файла:    maatriks_funktsioonidega.c
Автор:
Создано:
Обновлено:

Описание:    Создается матрица с заданными размерами
              Матрица заполняется случайными числами
              Содержимое матрицы выводится на экран
*/

#include <stdio.h>
#include <time.h>

```

```

#include <stdlib.h>

#define RIDU 5
#define VEERGE 5

//Прототипы функций
void taidaMaatriks(int, int, int [] [VEERGE], int);
void valjastaMaatriks(int, int, int [] [VEERGE]);
int juhuslikArv(int);

int main (void)
{
//декларирование матрицы, для которой
// RIDU - число рядов, VEERGE - число столбцов
    int maatriks[RIDU][VEERGE];

//инициализация генератора случайных чисел
    srand(time(0)) ;

    //вызов функции taidaMaatriks
    taidaMaatriks(RIDU, VEERGE, maatriks,20);

    //вызов функции valjastaMaatriks
    valjastaMaatriks(RIDU, VEERGE, maatriks);

//при необходимости добавляем следующие строки
//system ("PAUSE");
// или
// getchar(); getchar();
    return 0;
}

/* taidaMaatriks

Функция для заполнения матрицы случайными числами
Входные параметры:
    argRidu: целое число, число рядов матрицы
    argVeerge: целое число, число столбцов матрицы
    argMaatriks: ссылка на матрицу
    vaartus: целое число, матрица заполняется случайными числами в
промежутке 1..vaartus
    Возвращаемое значение: отсутствует
*/
void taidaMaatriks(int argRidu, int argVeerge, int argMaatriks[][VEERGE], int
vaartus)
{
    int i,j;
    for(i=0; i<argRidu; i++)
    {
        for(j=0; j<argVeerge; j++)
        {
            argMaatriks[i][j]=juhuslikArv(vaartus);
        }
    }
    return;
}

// данная функция генерирует значения в промежутке 1.. maxVaartus
int juhuslikArv(int maxVaartus)
{
    return rand()%maxVaartus + 1;
}

```

```

/* valjastaMaatriks

Функция для вывода матрицы на экран
Входные параметры:
    argRidu: целое число, число рядов матрицы
    argVeerge: целое число, число столбцов матрицы
    argMaatriks: ссылка на матрицу
Возвращаемое значение: отсутствует
*/
void valjastaMaatriks(int argRidu, int argVeerge, int argMaatriks[][VEERGE])
{
    int i,j;
    i=j=0;
    while (i<argRidu)
    {
        while (j<argVeerge)
        {
            printf("%4d", argMaatriks[i][j]);
            j++;
        }
        i++; j=0;
        printf("\n");
    }
    return;
}

```

- В ходе работы программа выводит следующую информацию на экран (выводится матрица, состоящая из 5 строк (рядов) и 5 столбцов):

```

c:\ggg\gg1\Debug\gg1.exe
13  4  8  5  9
 2 14  2 11  4
 8  9 18 13  5
 4  5 20  3  7
12  7  5  9  4
Press any key to continue_

```

Задание

Дополнить программу `maatriks_funktsioonidega.c` следующим образом:

- Создать функцию, которая находит и выводит на экран наибольший элемент матрицы
- Создать функцию, которая находит и выводит на экран наименьший элемент матрицы
- Создать функции (2), которые находят и выводят наибольший и наименьший элементы главной диагонали

• Упражнение 2

Следующая функция сортирует массив целых чисел по методу Шелла. Основная идея сортировки по Шеллу заключается в том, что сначала сравниваются удаленные элементы, а не смежные, как в обычном методе сортировки. Это приводит к быстрому устранению большей части неупорядоченности и сокращает последующую работу. Интервал между

элементами постепенно сокращается до единицы, когда сортировка фактически превращается в метод перестановки соседних элементов.

```
void shell(int v[], int n)
/* sort v[0]...v[n-1] into increasing order */
{
int gap, i, j, temp;

for (gap = n/2; gap > 0; gap /= 2)
    for (i = gap; i < n; i++)
        for (j=i-gap; j>=0 && v[j]>v[j+gap]; j-=gap)
            {
                temp = v[j];
                v[j] = v[j+gap];
                v[j+gap] = temp;
            }
}
```

Здесь имеются три вложенных цикла. Самый внешний цикл управляет интервалом между сравниваемыми элементами, уменьшая его от $n/2$ вдвое при каждом проходе, пока он не станет равным нулю. Средний цикл сравнивает каждую пару элементов, разделенных на величину интервала; самый внутренний цикл переставляет любую неупорядоченную пару. Так как интервал, в конце концов, сводится к единице, все элементы в результате упорядочиваются правильно. Отметим, что в силу общности конструкции `for` внешний цикл укладывается в ту же самую форму, что и остальные, хотя он и не является арифметической прогрессией. (Б. Керниган, Д. Ричи. Язык С.)

Задание

Исследовать сортировку по методу Шелла. При оформлении отчета привести пример сортировки (рисунок).

Дополнить программу `maatriks_funktsioonidega.c` следующим образом:

- Добавить функцию, которая, используя сортировку Шелла, сортирует и выводит на экран вектор с размерностью 9 и с начальной инициализацией элементов (произвольно задать элементы)
- Создать функцию, которая, используя сортировку Шелла, сортирует строки матрицы и выводит матрицу на экран

Отчет оформляется в MS Word (или в другом текстовом редакторе), представляется в распечатанном виде. Представляемые программы должны быть прокомментированы.

Использованы материалы:

- **Громов, Титаренко.** Программирование на языке С.
- **Керниган Б., Ритчи Д. Язык С.**
- **Материалы Helena Kruus (MSc)**
- **Материалы Tarmo Robal (MSc)**

Марина Брик
Составлено: 20.11.2007
Обновлено: 28.10.2008
Обновлено: 18.11.2010