# Тема 7. Одномерные массивы

**Т7.1** Для работы с множествами однотипных элементов в Си введено понятие массив. Такие множества элементов обозначаются какими-то именами, а для идентификации их отдельных элементов используются индексы. Такой подход позволяет сокращать объем текста программ и работать с массивами посредством операторов цикла.

**Задача.** В одномерном массиве из 100 элементов вещественного типа определить максимальное значение. Значения элементов массива вводим с клавиатуры. Наибольшее значение определяем с помощью стандартного алгоритма. Вначале полагаем, что максимальным элементом является элемент с нулевым индексом, т.е. первый элемент массива. Затем в цикле, сопоставляя его значение с другими, проверяем и уточняем значение максимального элемента.

#include <stdio.h>

int main(){

float a[100]; //объявление одномерного

//массива размерности 100

int i,n;float max;

printf("\nВведите элементы массива\n");

for(i=0;i<100;i=i+1)

{printf("\nВведите элемент a[%d] ",i);

scanf("%f",&a[i]);//чтение элементов

//массива с клавиатуры

};

printf("\nИсходный массив \n");

for(i=0;i<100;i++)

printf("%f ",a[i]);//вывод элементов

//массива на экран

max=a[0];

for(i=1;i<100;i++)

if(max<a[i])max=a[i];

printf("\nmax=%f",max);

return 1;

}

**Т7.2** Программа вычисляет среднее и дисперсию для

*n* значений элементов одномерного массива, вводимых с клавиатуры. Первоначально резервируется место в памяти для 100 элементов. Фактическое количество элементов *n* задается в цикле while.Причем условие продолжения цикла всегда истинное. Выход же из бесконечного цикла осуществляется с помощью оператора break .

#include<stdio.h>

int main()

{float x[100],s,a,b,d,r;

int i,j,n;

while(1)

{

printf("\nВведите n:\n");scanf("%d",&n);

if(n>0 && n <= 100) break;

printf("Ошибочное значение n!\n");

}

printf("Введите элементы массива x\n");

for(b=0.0,i=0;i<n;i++)

{printf("x[%d]=",i);scanf("%f",&x[i]);

b+=x[i];

}

b/=n;

printf("\nb=%f \n",b); //Вывод среднего

for(j=0,d=0.0;j<n;j++)

{a=x[j]-b;

d+=a\*a;}

d/=n;

printf("\nd=%f \n",d); //Вывод дисперсии

return 1;

**Т7.3** Ещё одна разновидность обработки одномерных массивов – сортировка, т.е. упорядочение их значений в некотором порядке. Программа данного примера сортирует элементы одномерного массива в порядке возрастания. Реализован алгоритм прямого упорядочения.

#include<stdio.h>

int main()

{double a[100],h;

int i,j,n;

while(1)

{

printf("\nВведите n:\n");scanf("%d",&n);

if(n>0 && n <= 100) break;

printf("Ошибочное значение n!\n");

}

printf("Введите элементы массива a\n");

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

{printf("a[%d]=",i);scanf("%lf",&a[i]);

}

for(i=0;i<n-1;i++)

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

if(a[i]>a[j])

{h=a[i]; a[i]=a[j]; a[j]=h;

}

printf("Упорядоченный массив:\n");

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

printf("a[%d]=%lf\n",i,a[i]);

return 1;

}

# Тема 8. Двумерные массивы

**Т8.1** Двумерный массив – это массив данных имеющий двойную индексацию. Такой тип данных применяют для обработки табличных данных, таких как числовые матрицы. За номера строк матрицы отвечает первый индекс массива, а за столбцы – второй, причем значения индексов начинаются с нулевых значений.

**Задача.** Вычислить суммы элементов столбцов целочисленной матрицы . Первоначально объем памяти резервируется для матрицы размерности *Lim* х *Lim.* Причем значение константы *Lim* задается с помощью директивы #define. Фактическая размерность матрицы читается с клавиатуры с помощью циклов, в которых проверяется корректность вводимых значений переменных *n* и *m*. После ввода с клавиатуры значений матрицы осуществляется её вывод на экран в табличном виде, т.е. построчно. После вывода каждой строки матрицы курсор переводится на следующую строку экрана.

#include <stdio.h>

#define Lim 30

int main(){

int i,j,n,m,sum,a[Lim][Lim];

printf("\Ввод размерности матрицы\n");

do{

printf("Введите 0<n<=%d ",Lim);

scanf("%d",&n);

}while(n<=0 || n>Lim);

do{

printf("Введите 0<m<=%d ",Lim);

scanf("%d",&m);

}while(m<=0 || m>Lim);

printf("Заполнение матрицы\n");

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

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

printf("Введите элемент %d %d ",i,j);

scanf("%d",&a[i][j]);

}

printf("Вывод матрицы а(n,m)\n");

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

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

printf("%2d ",a[i][j]);

printf("\n");

}

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

sum=0;

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

sum=sum+a[i][j];

printf("сумма столбца %d = %d\n",j,sum);

}

return 1;

}