

# Массивы

**Массив** – это совокупность данных, которая обладает следующими свойствами:

- все элементы массива имеют один и тот же тип;
- массив имеет одно имя для всех элементов;
- доступ к конкретному элементу массива осуществляется по индексу (индексам).

*<тип> <имя> [<константное выражение>]*

*<тип> <имя> [ ]*

*<тип> <имя> [<KB>][<KB>]...*

*// Одномерный массив из 10 целых чисел*

*// Индексы меняются от 0 до 9*

```
int    x[10];
```

*// Двумерный массив вещественных  
чисел*

*// из 2 строк и 10 столбцов*

```
double y[2][10];
```

```
int    a[3]    = {0, 1, 2};
```

```
double b[5]    = {0.1, 0.2, 0.3};
```

```
int    c[ ]    = {1, 2, 4, 8, 16};
```

```
int    d[2][3] = {{0, 1, 2}, {3, 4, 5}};
```

```
int    e[3]    = {0, 1, 2, 3};
```

Доступ к элементу одномерного массива:

$$x[i]$$

Доступ к элементу двумерного массива:

$$y[i][j]$$

$$x[i] \equiv *(x + i)$$

```
for (int i = 0;  
    i < n;  
    i++)  
    <тело цикла>
```

```
int f(..., int x[], ...) { ... }
```

```
int f(..., int *x, ...) { ... }
```

```
void main()
```

```
{ int a[10];
```

```
    ...
```

```
    f(..., a, ...);
```

```
}
```

```
int g(..., int x[][10], ...) { ... }
```



```
int a[10], n;

printf("Введите количество элементов массива
      (от 0 до 10): ");
scanf("%d", &n);
if (n < 0 || n > 10)
{   printf("Количество элементов массива
        должно быть от 0 до 10!\n");
    return;
}
for (int i = 0; i < n; i++)
    scanf("%d", &a[i]);    // scanf("%d", a + i)

for (int i = 0; i < n; i++)
    printf("a[%d] = %3d\n", i + 1, a[i]);
```

```
#include <stdio>

const int nmax = 10;

int ArrayInput(int *n, double x[], char *fname);
double Sum(double x[], int n);

void main(int argc, char *argv[])
{ double a[nmax], b[nmax], c[nmax];
  double sa, sb, sc, max;
  int na, nb, nc;

  if (argc < 4)
  { printf("Недостаточно параметров!\n");
    return;
  }
  if (!ArrayInput(&na, a, argv[1]))
    return;
  if (!ArrayInput(&nb, b, argv[2]))
    return;
  if (!ArrayInput(&nc, c, argv[3]))
    return;
```

```
sa = Sum(a, na);
sb = Sum(b, nb);
sc = Sum(c, nc);

max = sa;
if (sb > max) max = sb;
if (sc > max) max = sc;

if (sa == max)
    printf("Массив А имеет максимальную сумму элементов: %9.31f\n",
        max);
if (sb == max)
    printf("Массив В имеет максимальную сумму элементов: %9.31f\n",
        max);
if (sc == max)
    printf("Массив С имеет максимальную сумму элементов: %9.31f\n",
        max);
}
```

```
double Sum(double x[], int n)
{ double s = 0;

  for (int i = 0; i < n; i++)
    s += x[i];
  return s;
}
```

```
int ArrayInput(int *n, double x[], char *fname);
{ FILE *file;

    if ((file = fopen(fname, "r")) == NULL)
    { printf("Невозможно открыть файл '%s'\n", fname);
      return 0;
    }
    if (fscanf(file, "%d", n) < 1)
    { printf ("Ошибка чтения из файла '%s'\n", fname);
      fclose(file);
      return 0;
    }
    if (*n < 0 || *n > nmax)
    { printf("Кол-во эл-тов масс. должно быть от 0 до %d! (файл '%s')\n", nmax,
fname);
      fclose(file);
      return 0;
    }
    for (int i = 0; i < *n; i++)
    if (fscanf(file, "%d", &x[i]) < 1)
    { printf ("Ошибка чтения из файла '%s'\n", fname);
      fclose(file);
      return 0;
    }
    fclose(file);
    return 1;
}
```

```
#include <stdio>
const int nmax = 10;

void Zeros(double x[][nmax], int m, int n, int z[]);

void main(int argc, char *argv[])
{ double a[nmax][nmax];
  int m, n, z[nmax];
  FILE *file;

  if (argc < 2)
    { printf("Недостаточно параметров!\n"); return; }
  if ((file = fopen(argv[1], "r")) == NULL)
    { printf("Невозможно открыть файл '%s'\n", argv[1]); return; }

  if (fscanf(file, "%d%d", &m, &n) < 2)
    { printf ("Ошибка чтения из файла '%s'\n", argv[1]);
      fclose(file);
      return;
    }
  if (m < 0 || m > nmax || n < 0 || n > nmax)
    { printf("К-во с. и ст. матрицы должна быть от 1 до %d!\n", nmax);
      return;
    }
}
```

```
for (int i = 0; i < m; i++)  
    for (int j = 0; j < n; j++)  
        if (fscanf(file, "%lf", &a[i][j]) < 1)  
            { printf("Произошла ошибка чтения из файла '%s'\n", argv[1]);  
              return;  
            }  
}
```

```
Zeros(a, m, n, z);
```

```
for (int i = 0; i < m; i++)  
    if (z[i])  
        printf("В %3d строке есть нулевые элементы\n", i + 1);  
}
```

```
void Zeros(double x[][nmax], int m, int n, int z[])  
{ int i, j;  
  
    for (i = 0; i < m; i++)  
        for (z[i] = 0, j = 0; j < n; j++)  
            if (x[i][j] == 0) // if (fabs(x[i][j]) < 0.0001)  
                { z[i] = 1; break; }  
}
```

```
#include <stdio>

const int nmax = 10;

int Zeros(double x[], int n);

void main(int argc, char *argv[])
{ double a[nmax][nmax];
  int m, n;
  FILE *file;

  . . .      // Ввод матрицы
  for (int i = 0; i < m; i++)
    if (Zeros(a[i], n))
      printf("В %3d строке есть нулевые элементы\n", i + 1);
}

int Zeros(double x[], int n)
{ for (int j = 0; j < n; j++)
  if (x[j] == 0)
    return 1;
  return 0;
}
```



```
#include <stdio>
```

```
double Sum(double *x, int m, int n);
```

```
void main(int argc, char *argv[])
```

```
{ const int na = 4, mb = 3, nb = 5;
```

```
  double a[na][na], b[mb][nb];
```

```
  double sa, sb;
```

```
  FILE *file;
```

```
  . . .          // Ввод матриц
```

```
  sa = Sum(a[0], na, na);
```

```
  sb = Sum(reinterpret_cast<double *>(b), mb, nb);
```

```
  printf("SumA = %6.21f\nSumB = %6.21f\n", sa, sb);
```

```
}
```

```
double Sum(double *x, int m, int n)
```

```
{ double s = 0;
```

```
  for (int i = 0; i < m; i++)
```

```
    for (int j = 0; j < n; j++)
```

```
      s += x[i * n + j];
```

```
  return s;
```

```
}
```