

## Практическое занятие

## Передача массивов как параметров в Функции

Имеется три способа объявления формального параметра функции определяющего массив.

1. Он может быть объявлен как массив:

```
#include <stdio.h>
void display(int num[10])
    { int i; for (i=0; i<10; i++) printf ("%d", num[i]); }
int main (void) /* ВЫВОД ЧИСЕЛ */
{ int t [10], i;
  for (i=0; i<10; ++i) t[i]=i;
  display(t);
  return 0;
}
```

2. Следующий способ объявляет параметр как безразмерный массив

```
void display(int num[])
    { int i; for (i=0; i<10; i++) printf("%d ", num[i]); }
```

где num объявлен как целочисленный массив неизвестного размера. Поскольку Си не предоставляет проверку границ массива, настоящий размер массива не имеет никакого отношения к параметру (но, естественно, не к программе).

3. Последний способ, которым может быть объявлен num, - это наиболее типичный способ, применяемый при написании профессиональных программ, - через указатель, т.е. переменную хранящую адрес первого байта массива в оперативной памяти.

# Массивы

## Практическое занятие

### Выполнить C \ C++

Ввести с клавиатуры двумерный массив целых чисел размерностью 5x5.

Вывести на печать разность сумм элементов главной и побочной диагоналей матрицы.

Главная диагональ матрицы проходит от левого верхнего угла (элемента) матрицы к правому нижнему.

Побочная диагональ матрицы проходит от правого верхнего угла (элемента) матрицы к левому нижнему.

Оформить вычисление суммы элементов диагонали как функцию.

Ввести с клавиатуры двумерный массив целых чисел размерностью 2x2 .

Вывести на печать определитель матрицы 2-го порядка.

Определитель матрицы, детерминант [determinant] – число, соответствующее матрице и полученное путем ее преобразования по определенному правилу.

Для матрицы 2-го порядка – это:

$$\det A = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}$$

Оформить вычисление определителя через процедуру.

# МАССИВЫ. Сортировка

C /C++

Создать функцию  
выборочной сортировки

C

Практическое занятие: сортировка и слияние двух массивов (на языке C)  
используя функции выборочной сортировки и слияния целочисленных массивов

```
// Сортировка мас. целых чисел выборочн. методом
#include <stdio.h>
#include <conio.h>
#define sz 5 // размерность массива
void main ()
{ int a[sz]; // массив целых чисел
  int i; // № элем., от которого ведется поиск мин. элем.
  int min; // № мин. элем. в части мас. от i до конца мас.
  int j; // № элемента сравниваемого с мин.
  int buf; // буфер, исп. при обмене элементов массива
  int k; // индекс для ввода и вывода
  printf ("\nВведите в одной строке %i", sz);
  printf (" целых чисел и нажмите Enter\n");
  printf ("-> ");
  for (k=0; k<sz; k++) scanf ("%i", &a[k]);
  // Сортировка
  for (i = 0; i < sz-1; i++)
  { // Поиск мин. элем. в части мас. от a[i] до a[sz]
    min = i; for (j = i+1; j < sz; j++)
      if (a[j] < a[min]) min = j;
  // Меняем местами a[min] и a[i]
    buf = a[i]; a[i] = a[min]; a[min] = buf;
  }
  // Цикл сортировки закончен
  // Вывод отсортированного массива
  printf ("Отсортированный массив\n");
  for (k = 0; k<sz; k++) printf ("%i ", a[k]);
}
```

Формат описания Функции:

[класс] <возвр\_тип>  
<имя\_функции> [(тип1  
имя\_формального\_параметра1, ..., типN имя\_формального\_параметраN)] [throw  
(исключения)]  
{  
<тело\_функции >  
}

Формат вызова Функции:

<Имя\_функции>  
[(фактич\_параметр1, ..., фактич\_параметрN)];

# МАССИВЫ. Сортировка

C \ C++

## Создать функцию выборочной сортировки

Практическое занятие: сортировка и слияние двух массивов (на языке C)  
используя функции выборочной сортировки и слияния целочисленных массивов

```
// Сортировка мас. целых чисел выборочн. методом
#include <stdio.h>
#include <conio.h>
#define sz 5 // размерность массива
void main ()
{ int a[sz]; // массив целых чисел
  int i; // № элем., от которого ведется поиск мин. элем.
  int min; // № мин. элем. в части мас. от i до конца мас.
  int j; // № элемента сравниваемого с мин.
  int buf; // буфер, исп. при обмене элементов массива
  int k; // индекс для ввода и вывода
  printf ("\nВведите в одной строке %i", sz);
  printf (" целых чисел и нажмите Enter\n");
  printf ("-> ");
  for (k=0; k<sz; k++) scanf ("%i", &a[k]);
  // Сортировка
  for (i = 0; i < sz-1; i++)
  { // Поиск мин. элем. в части мас. от a[i] до a[sz]
    min = i; for (j = i+1; j < sz; j++)
      if (a[j] < a[min]) min = j;
  // Меняем местами a[min] и a[i]
    buf = a[i]; a[i] = a[min]; a[min] = buf;
  }
  // Цикл сортировки закончен
  // Вывод отсортированного массива
  printf ("Отсортированный массив\n");
  for (k = 0; k<sz; k++) printf ("%i ", a[k]);
}
```

```
void direct_sort (int a[sz])
// Функция выборочной сортировки массива
целых чисел по возрастанию
{
  int i; // № элем., от которого ведется поиск мин. элем.
  int numin; // № минимального элемента
  int j; // № элемента сравниваемого с минимальным
  int buf; // буфер, исп. при обмене элементов массива

  for (i = 0; i < sz-1; i++)
  {
    // Поиск мин. элемента в части массива от a[i] до a[sz-1]
    numin = i;
    for (j = i+1; j < sz; j++)
      if (a[j] < a[numin])
        numin = j;
    // Меняем местами a[numin] и a[i]
    buf = a[i];
    a[i] = a[numin];
    a[numin] = buf;
  } // цикл сортировки закончен
}
```

# МАССИВЫ. Сортировка

## Создать функцию слияния отсортированных массивов

C \ C++

Практическое занятие: сортировка и слияние двух массивов (на языке C)  
используя функции выборочной сортировки и слияния целочисленных массивов

```
#include <stdio.h> // Слияние двух упорядоченных
                    массивов
#include <conio.h>
#define SZ 5        //Размер исходных массивов
void main() {
    int a[SZ], b[SZ]; // исходные массивы
    int c[SZ*2];     // массив-результат
    int k,i,m;       // индексы массивов a, b и c
    printf ("Слияние двух упорядоченных массивов в
    один,\n");
    printf ("ввод элементов через пробел, завершение:
    Enter\n");
    printf ("\nВведите первый массив %i -> ", SZ);
    for (k=0; k<SZ; k++) scanf ("%i", &a[k]);
    printf ("Введите второй массив %i -> ", SZ);
    for (i=0; i<SZ; i++) scanf ("%i", &b[i]);
    k=i=m=0;
    do { if (a[k] < b[i]) c[m++] = a[k++];
        else if (a[k] > b[i]) c[m++] = b[i++];
        else { c[m++] = a[k++]; c[m++] = b[i++]; } }
    while ((k < SZ) && (i < SZ));
    while (k < SZ) /* есть элем. массива А не переписанные
    в C */
        c[m++] = a[k++];
    while (i < SZ) /* есть элементы В не переписанные в
    C */
        c[m++] = b[i++];
    printf("Массив-результат: \n");
    for (i=0; i<2*SZ; i++) printf ("%i ", c[i]);
    printf("\nДля завершения работы нажмите Enter\n");
    getch(); }
```

Формат описания Функции:

[класс] <возвр\_тип>  
<имя\_функции> [(тип1  
имя\_формального\_парамет-  
ра1, ..., типN имя\_формаль-  
ного\_параметраN)] [throw  
(исключения)]  
{  
<тело\_функции >  
}

Формат вызова Функции:

<Имя\_функции>  
([фактич\_параметр1, ...,  
фактич\_параметрN]);

# МАССИВЫ. Сортировка

C \ C++

## Создать функцию слияния отсортированных массивов

Практическое занятие: сортировка и слияние двух массивов (на языке C)  
используя функции выборочной сортировки и слияния целочисленных массивов

```
#include <stdio.h> // Слияние двух упорядоченных
                    массивов
#include <conio.h>
#define SZ 5        //Размер исходных массивов
void main() {
    int a[SZ], b[SZ]; // исходные массивы
    int c[SZ*2];     // массив-результат
    int k,i,m;       // индексы массивов a, b и c
    printf ("Слияние двух упорядоченных массивов в
    один,\n");
    printf ("ввод элементов через пробел, завершение:
    Enter\n");
    printf ("\nВведите первый массив %i -> ", SZ);
    for (k=0; k<SZ; k++) scanf ("%i", &a[k]);
    printf ("Введите второй массив %i -> ", SZ);
    for (i=0; i<SZ; i++) scanf ("%i", &b[i]);
    k=i=m=0;
    do { if (a[k] < b[i]) c[m++] = a[k++];
        else if (a[k] > b[i]) c[m++] = b[i++];
        else { c[m++] = a[k++]; c[m++] = b[i++]; } }
    while ((k < SZ) && (i < SZ));
    while (k < SZ) /* есть элем. массива A не переписанные
    в C */
        c[m++] = a[k++];
    while (i < SZ) /* есть элементы B не переписанные в
    C */
        c[m++] = b[i++];
    printf("Массив-результат: \n");
    for (i=0; i<2*SZ; i++) printf ("%i ", c[i]);
    printf("\nДля завершения работы нажмите Enter\n");
    getch(); }
```

```
void mas_split (int a[sz], int b[sz], int c[sz*2])
/* Функция слияния двух упорядоченных по
возрастанию массивов целых чисел в один */
{
    int k,i,m; // индексы массивов a, b и c
    k=i=m=0;
    do { if (a[k] < b[i])
        c[m++] = a[k++];
        else if (a[k] > b[i])
            c[m++] = b[i++];
        else
            { c[m++] = a[k++];
              c[m++] = b[i++]; }
    }
    while ((k < sz) && (i < sz));
    while (k < sz) /* есть элем. мас. a не переписанные в c
    c[m++] = a[k++];
    while (i < sz) /* есть элементы b не переписанные в c
    c[m++] = b[i++];
}
```

# МАССИВЫ. Сортировка

C \ C++

Практическое занятие: сортировка и слияние двух массивов (на языке C)  
используя функции выборочной сортировки и слияния целочисленных массивов

Функции:

```
void direct_sort (int a[sz])
// Функция выборочной сортировки массива
целых чисел по возрастанию
{
int i; // № элем., от которого ведется поиск мин. элем.
int numin; // № минимального элемента
int j; // № элемента сравниваемого с минимальным
int buf; // буфер, исп. при обмене элементов массива

for (i = 0; i < sz-1; i++)
{
// Поиск мин. элемента в части массива от a[i] до a[sz-1]
numin = i;
for (j = i+1; j < sz; j++)
if (a[j] < a[numin])
numin = j;
// Меняем местами a[numin] и a[i]
buf = a[i];
a[i] = a[numin];
a[numin] = buf;
} // цикл сортировки закончен
}
```

```
void mas_split (int a[sz], int b[sz], int c[sz*2])
/* Функция слияния двух упорядоченных по
возрастанию массивов целых чисел в один */
{
int k,i,m; // индексы массивов a, b и c
k=i=m=0;
do { if (a[k] < b[i])
c[m++] = a[k++];
else if (a[k] > b[i])
c[m++] = b[i++];
else
{ c[m++] = a[k++];
c[m++] = b[i++];}
}
while ((k < sz) && (i < sz));
while (k < sz) // есть элем. мас. a не переписанные в c
c[m++] = a[k++];
while (i < sz) // есть элементы b не переписанные в c
c[m++] = b[i++];
}
```

C \ C++

# МАССИВЫ.

## Сортировка

Практическое занятие: сортировка и слияние двух массивов (на языке C)  
используя функции выборочной сортировки и слияния целочисленных массивов

Головная программа:

```
void main()
/* Ввод двух целочисленных массивов, сортировка обоих массивов и слияние этих массивов в третий массив */
{   int a[sz], b[sz]; // исходные массивы целых чисел
    int c[sz*2];      // массив-результат
    int d, l, n;      // индексы массивов A, B и C
    srand(time(NULL));
    printf ("Сортировка и слияние двух упорядоченных по возрастанию массивов в один,\n");
    // Ввод первого массива целых чисел
    for (d = 0; d < sz; d++)  a[d] = rand() % 100; // как вариант: scanf ("%i", &a[d]);
    // Ввод второго массива целых чисел
    for (l = 0; l < sz; l++)   b[l] = rand() % 100; // как вариант: scanf ("%i", &b[l]);
    direct_sort (a);
    // Вывод отсортированного первого массива
    printf ("\n Отсортированный первый массив -> ");
    for (d = 0; d < sz; d++)   printf ("%i ", a[d]);
    direct_sort (b);
    // Вывод отсортированного второго массива
    printf ("\n Отсортированный второй массив -> ");
    for (l = 0; l < sz; l++)   printf ("%i ", b[l]);
    mas_split (a, b, c);
    // Вывод результата слияния двух массивов
    printf ("\n Массив-результат: \n");
    for (n = 0; n < 2*sz; n++) printf ("%i ", c[n]); }
```

Для работы функции **srand** надо подключать библиотеку **time.h**;  
функция **rand()** находится в библиотеке **stdlib.h**