

## 1. 一维数组

### 1.1 数组的定义

数组的定义方式和变量类似。

```
#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int a[10], b[10];
    float f[33];
    double d[123];
    char c[21];

    return 0;
}
```

### 1.2 数组的初始化

在main函数内部，未初始化的数组中的元素是随机的。

```
#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int a[3] = {0, 1, 2};           // 含有3个元素的数组，元素分别是0, 1, 2
    int b[] = {0, 1, 1};          // 维度是3的数组
    int c[5] = {0, 1, 2};        // 等价于c[] = {0, 1, 2, 0, 0}
    char d[3] = {'a', 'b', 'c'}; // 字符数组的初始化

    return 0;
}
```

### 1.3 访问数组元素

通过下标访问数组。

```

#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int a[3] = {0, 1, 2}; // 数组下标从0开始

    cout << a[0] << ' ' << a[1] << ' ' << a[2] << endl;

    a[0] = 5;

    cout << a[0] << endl;

    return 0;
}

```

练习题1：使用数组实现求斐波那契数列的第 N 项。

```

#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int n;
    int f[100];

    cin >> n;

    f[0] = 0, f[1] = 1;
    for (int i = 2; i <= n; i ++ ) f[i] = f[i - 1] + f[i - 2];

    cout << f[n] << endl;

    return 0;
}

```

练习题2：输入一个 n，再输入 n 个整数。将这 n 个整数逆序输出。

```

#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int n;
    int a[100];

    cin >> n;
    for (int i = 0; i < n; i ++ ) cin >> a[i];
    for (int i = n - 1; i >= 0; i -- ) cout << a[i] << ' ';
    cout << endl;

    return 0;
}

```

练习题3：输入一个  $n$ ，再输入  $n$  个整数。将这个数组顺时针旋转  $k$  ( $k \leq n$ ) 次，最后将结果输出。  
 旋转一次是指：将最左边的数放到最右边。

```

#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int n, k;
    int a[100];

    cin >> n >> k;
    for (int i = 0; i < n; i ++ ) cin >> a[i];

    reverse(a, a + k);
    reverse(a + k, a + n);
    reverse(a, a + n);

    for (int i = 0; i < n; i ++ ) cout << a[i] << ' ';
    cout << endl;

    return 0;
}

```

练习题4：输入  $n$  个数，将这  $n$  个数按从小到大的顺序输出。

```

#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int n;
    int a[100];

    cin >> n;
    for (int i = 0; i < n; i ++ ) cin >> a[i];

    for (int i = 0; i < n; i ++ )
        for (int j = i + 1; j < n; j ++ )
            if (a[i] > a[j])
                swap(a[i], a[j]);

    for (int i = 0; i < n; i ++ ) cout << a[i] << ' ';
    cout << endl;

    return 0;
}

```

练习题5：计算 22 的  $N$  次方。  $N \leq 10000 \leq 10000$

```

#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int a[10000], size = 1, n;
    a[0] = 1;

    cin >> n;
    while (n -- )
    {
        int t = 0;
        for (int i = 0; i < size; i ++ )
        {
            t += a[i] * 2;
            a[i] = t % 10;
            t /= 10;
        }
        if (t) a[size ++ ] = t;
    }

    for (int i = size - 1; i >= 0; i -- ) cout << a[i];
    cout << endl;

    return 0;
}

```

## 2. 多维数组

多维数组就是数组的数组。

```
int a[3][4]; // 大小为3的数组，每个元素是含有4个整数的数组。
```

```
int arr[10][20][30] = {0}; // 将所有元素初始化为0
// 大小为10的数组，它的每个元素是含有20个数组的数组
// 这些数组的元素是含有30个整数的数组
```

```
#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int b[3][4] = {           // 三个元素，每个元素都是大小为4的数组
        {0, 1, 2, 3},        // 第1行的初始值
        {4, 5, 6, 7},        // 第2行的初始值
        {8, 9, 10, 11}      // 第3行的初始值
    };

    return 0;
}
```

练习题：输入一个  $n \times m$  的矩阵，从左上角开始将其按回字形的顺序顺时针打印出来。

```
#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int n, m;
    int arr[50][50];

    cin >> n >> m;
    for (int i = 0; i < n; i ++ )
        for (int j = 0; j < m; j ++ )
            cin >> arr[i][j];

    bool st[50][50] = {false};
    int dx[4] = {-1, 0, 1, 0}, dy[4] = {0, 1, 0, -1};
    int d = 1, x = 0, y = 0;
    for (int i = 0; i < n * m; i ++ )
    {
        int a = x + dx[d], b = y + dy[d];
        if (a < 0 || a >= n || b < 0 || b >= m || st[a][b])
        {
            d = (d + 1) % 4;
            a = x + dx[d], b = y + dy[d];
        }
        cout << arr[x][y] << ' ';
        st[x][y] = true;
        x = a, y = b;
    }
    cout << endl;

    return 0;
}
```