

```

#include <cstdio>
#include <iostream>
#include <string>
#include <cstring>

using namespace std;

// 数组的初始化
void init_array();
// 数组的定义
void read_array();
// 输入整个数组
void scanf_array();
// 二维数组的定义和输入
void twod_array();
// 二维数组的大小
void size_twod_array();

int main(){
    // init_array();
    // read_array();
    // scanf_array();
    // twod_array();
    // size_twod_array();
    return 0;
}

void init_array(){
    // 数组的初始化
    int x[10] = {0}; // 初始化我这个数组所有的值为0
    for(int i = 0; i < 10; i++){
        printf("%d ", x[i]);
    }
}

void read_array(){
    /*
    这几种定义都可以
    int x[10] = {1,2,3,4,5,6,7,8,9,10};

```

```

    int x[100] = {1,2,3,4,5,6,7,8,9,10};
*/
int x[] = {1,2,3,4,5,6,7,8,9,10};

for(int i = 0; i < 10; i++){
    x[i] = 100;
}

for(int i = 0; i < 10; i++){
    cout << x[i] << endl;
}

// 数组的长度
// sizeof(x) x占用了多少的bytes 字节
cout << "x的大小(bytes): " << sizeof(x) << endl;
cout << "x[0]的大小: " << sizeof(x[0]) << endl;
// sizeof(int) 或者 sizeof(x[0])
cout << "x数组的长度: " << sizeof(x)/sizeof(int) << endl;
}

void scanf_array(){
    // 输入整个数组
    int x[10];
    for(int i = 0; i < 10; i++){
        // cin >> x[i];
        scanf("%d", &x[i]);
    }
    // 输出整个数组
    for(int i = 0; i < 10; i++){
        printf("%d ", x[i]);
    }
}

void twod_array(){
    // 二维数组的定义
    // int x[3][3];
    // int x[][3] = {{1,2,3}, {4,5,6}, {7,8,9}};
    int x[3][3] = {{1,2,3}, {4,5,6}, {7,8,9}};
}

```

```

// int x[][3] = {1,2,3,4,5,6,7,8,9,10,11}; // 推荐
// 二维数组的赋值
x[2][2] = 100;
// 二维数组的输入
for(int i = 0; i < 3; i++){ // 循环 m 次
    for(int j = 0; j < 3; j++){ // 循环 n 次
        /*
            x是变量: scanf("%d", &x);
            x是一维数组: scanf("%d", &x[i]);
            x是二维数组: scanf("%d", &x[i][j]);
        */
        scanf("%d", &x[i][j]);
    }
}
// 二维数组的打印
for(int i = 0; i < 3; i++){
    for(int j = 0; j < 3; j++){
        cout << x[i][j] << " ";
    }
    cout << endl;
}
}

void size_twod_array(){
    int x[3][3] = {{1,2,3}, {4,5,6}, {7,8,9}};
    // 二维数组的大小
    printf("x占用字节: %lu\n", sizeof(x)); // 9 个int 36
    printf("x[0]占用字节: %lu\n", sizeof(x[0])); // 12
    printf("x[0][0]占用字节: %lu\n", sizeof(x[0][0])); // 4
    printf("多少行(大数组里的小数组有多少个): %lu\n", sizeof(x) /
sizeof(x[0])); // 3 行
    printf("多少列(小数组里的存的int有多少个): %lu\n",
sizeof(x[0])/sizeof(x[0][0])); // 3 列

    // 二维数组的地址
    /*
        x: 0x16ceed04
        x[0]: 0x16ceed04
        x[1]: 0x16ceed10
    */
}

```

&x[1][0]: 0x16ceed10

总结 x x[0] &x[0][0] 存的就是第一个元素x[0][0]的地址

x[1] &x[1][0] 存的都是第一列第0行元素的地址

*/

```
cout << x << endl; // 0x16b816d04 代表地址 pointer
```

```
cout << x[0] << endl;
```

```
cout << x[1] << endl;
```

```
cout << &x[1][0] << endl;
```

```
}
```