day03

一、指针的介绍及定义

1.指针是什么?

> 指针其实就是普通的一个变量，不过他的值是一个内存的地址

2.声明和初始化

- *声明*

```
// 类型* 名字
    int* p;
// 设置成空指针
    int* p = NULL;
```

- *初始化*

```
int main(){
    // 初始化
    int age = 10;
    int* p = &age;

    int age2 = 20;
```

```
    int* p1;
    p1 = &age2;
    cout << &age << "=" << p << endl;
    cout << &age2 << "=" << p1 << endl;
    return 0;
}
// 运行结果:
0x61fd3c=0x61fd3c
0x61fd38=0x61fd38
```

二、指针的两种定义方法

1.第一种、栈内存的地址

```
// 栈内存的地址
int a = 10;
int* p = &a;
```

2.第二种、堆内存地址

```
// 堆内存地址
/*
 * 返回堆地址 = new 数据类型(数据)
 */
int *p2;
p2 = new int(10);
delete p2; // 释放p2内存
```

三、解引用和交换两个变量的值

1.解引用

```
/*
 * 指针的解引用
 */
int main(){
    int age = 10;
    int* p = &age;
    // 解引用其实就是获取指针指向的内存对应的值
    cout << *p << endl;
    *p = 28;
    cout << *p << endl;
    return 0;
}
// 运行结果:
10
28
```

2.交换两个变量的值

```
void swap(int* x;int* y){
    int tmp;
    &tmp = x;
    x = y;
    y = &tmp;
}
int main(){
    int a = 3;
    int b = 4;
    swap(a,b);
    cout << "a = " << a << endl;
    cout << "b = " << b << endl;
}
```

四、指针和数组的关系及运算

1.指针和数组的关系

```
/*
 * 指针和数组
 */
int main(){

    int score[]{10,32,54,87,95};
    int* p = score;
    cout << score << "=" << p << endl;
    // 取值
    cout << *(p+0) << "=" << p[0] << endl;
    cout << *(p+1) << "=" << p[1] << endl;
    cout << *(p+2) << "=" << p[2] << endl;
    return 0;
}
// 运行结果:
0x61fd30=0x61fd30
10=10
32=32
54=54
```

指针的运算：加数据，实则表示指针的偏移

2.指针的运算

`++` `-` `>` `<` `>=` `<=`都可以进行计算

3.指针和常量

```
int age = 18;
// 指向常量的指针
// 但可以修改指向
const int* p1 = &age;
// 常量指针
// 可以修改数据
int* const p2 = &age;
```

五、指针和函数

1.指针作为函数参数

普通数据传递

```
void swap(int* x;int* y){
    int tmp;
    &tmp = x;
    x = y;
    y = &tmp;
}
int main(){
    int a = 3;
    int b = 4;
    swap(a,b);
    cout << "a = " << a << endl;
    cout << "b = " << b << endl;
}
// 运行结果:
a = 4
b = 3
```

传递vector数据

```
/*
 * 函数传递指针
 */
void changeScore(vector<int>* vi){
    (*vi).at(2) = 88;
}
int main(){

    vector<int> score{10,20,30};
    score.push_back(40);
    changeScore(&score);
    cout << score.at(2) << endl;
    return 0;
}
// 运行结果:
```

```
16  88
```

2.指针作为函数返回值

```
1  int* getMax(int* a,int* b){
2      if(a > b){
3          return a;
4      }else{
5          return b;
6      }
7  }
```

永远不要返回局部变量的值

六、二级指针

指向指针的变量是二级指针

```
1  int age = 18;
2  int* p1 = &age;
3  int** p2 = &p1;
```

二级指针出现的地方式函数参数传递

```
1  void initPoint(int** a){
2      // 解引用
3      *a = new int();
4  }
5  int mian(){
6      int* p = NULL;
7      initPoint(&p);
8      // 修改值
9      *p = 55;
10 }
```

空指针不能进行解引用