

# 地址与指针

# 地址

- 计算机中保存变量的物理位置，用一串16进制的数字表示
- `scanf("%d", &a);` 输入一个整数保存到a变量所在的地址上
- `printf("%x", &a);` 输出a变量所在地址

# 指针

- 指针类型的变量储存的是地址
- 对指针可以进行偏移，遍历数据结构，取值，改变值等操作，由于直接跟地址打交道，速度比较快
- 一个指针类型的变量占用8个字节

```
int main() {  
    int* p;  
    int a = 10;  
    p = &a;  
    *p = 20;  
    cout << a << endl;  
    return 0;  
}
```

```
void change(int *p) {  
    *p = 20;  
}
```

```
int main() {  
    int a = 10;  
    int* p = &a;  
    change(p);  
    cout << a << endl;  
    return 0;  
}
```

```
void change(int *arr, int n) {
    for (int i = 0; i < n; i++) {
        arr[i] = -1;
    }
}

int main() {
    int a[10];
    memset(a, 0, sizeof(a));
    change(a, 10);
    for (int i = 0; i < 10; i++) {
        cout << a[i] << " ";
    }
    return 0;
}
```

```
void fun(int a[][110]) {
}
```

```
void combine(int* first1, int* last1, int* first2, int* last2, int* p) {  
    while (true) {  
        if (first1 == last1) {  
            copy(first2, last2, p);  
            return ;  
        }  
        if (first2 == last2) {  
            copy(first1, last1, p);  
            return ;  
        }  
        *p++ = (*first1 < *first2) ? *first1++ : *first2++;  
    }  
}
```

```
int main() {  
    int a[] = {1 ,3 ,5 ,7 ,8};  
    int b[] = {2, 4, 6, 9, 10};  
    int c[10];  
    combine(a, a + 5, b, b + 5, c);  
    for (int i = 0; i < 10; i++) {  
        cout << c[i] << " ";  
    }  
    return 0;  
}
```

# 引用

- 引用是变量的别名，底层也使用指针实现
- 改变引用变量的值就相当于改变原变量的值
- 不存在空引用，引用必须链接到一块合法的内存
- 引用必须在创建时被初始化，创建后不能指向其他对象

```
int main() {  
    int b = 10;  
    int &a = b;  
    a = 20;  
    cout << b << endl;  
    return 0;  
}
```

```
void update(int &x, int y) {  
    if (y > x) {  
        x = y;  
    }  
}
```



# 实战训练

- 如果有编译错误，修复它
- 如果无编译错误，会输出什么

```
int &fun()
{
    static int x = 10;
    return x;
}
int main()
{
    fun() = 30;
    cout << fun();
    return 0;
}
```

```
int fun(int &x)
{
    return x;
}
int main()
{
    cout << fun(10);
    return 0;
}
```

```
void swap(char * &str1, char * &str2)
{
    char *temp = str1;
    str1 = str2;
    str2 = temp;
}

int main()
{
    char *str1 = "GEEKS";
    char *str2 = "FOR GEEKS";
    swap(str1, str2);
    cout<<"str1 is "<<str1<<endl;
    cout<<"str2 is "<<str2<<endl;
    return 0;
}
```

```
int main()
{
    int x = 10;
    int *ptr = &x;
    int &*ptr1 = ptr;
    cout << ptr1 << endl;
}
```

```
#include<iostream>
using namespace std;

int main()
{
    int *ptr = NULL;
    int &ref = *ptr;
    cout << ref;
}
```

```
#include<iostream>
using namespace std;

int &fun()
{
    int x = 10;
    return x;
}
int main()
{
    fun() = 30;
    cout << fun();
    return 0;
}
```