什么是函数指针？

- 函数指针(Function Pointer)就是指向函数的指针变量

  数据类型 (*指针变量名)(形参列表);

- 例如，若有函数原型为：

  ```
  int Fun(int a, int b);
  ```

- 则可定义函数指针

  ```
  int (*f)(int, int);
  ```

- 令f = Fun;就是让f指向函数Fun()
  - 编译器将不带()的函数名解释为该函数的入口地址
  - 函数指针变量存储的是函数在内存中的入口地址

阅读：课本``9.6 函数指针及其应用"
前两个自然段

函数指针的定义

- 而若有函数原型为：

```
float Fun(float a, float b);
```

- 则需定义函数指针

```
float (*f)(float, float);
```

- 令`f = Fun;`
- 定义时的参数类型与指向的函数参数类型不匹配

```
float (*f)(int, int); //错误
float (*f)();         //不建议
```

定义函数指针时的常见错误

```
int (*f)(int, int);
```

- 忘了写前一个()

```
int *f(int, int);
```

  - 声明了一个函数名为f、返回值是整型指针类型的函数

- 忘了写后一个()

```
int (*f);
```

  - 定义了一个整型指针变量

函数指针有什么用？

```c
#include <stdio.h>
int Max(int x, int y);
int main()
{
    int a, b, result;
    int (*f)(int, int);
    scanf("%d,%d", &a, &b);
    f = Max;
    result = (*f)(a, b);
    printf("%d\n", result);
    return 0;
}
int Max(int x, int y)
{
    printf("max=");
    return  x > y? x : y;
}
```

舍近求远，多此一举？

```c
#include <stdio.h>
int Max(int x, int y);
int main()
{
    int a, b, result;
    scanf("%d,%d", &a, &b);
    result = Max(a, b);
    printf("%d\n", result);
    return 0;
}
int Max(int x, int y)
{
    printf("max=");
    return  x > y? x : y;
}
```

函数指针有什么用？

```
#include <stdio.h>
void Fun(int x, int y, int (*f)(int, int));
int Max(int x, int y);
int Min(int x, int y);
int Add(int x, int y);
int main()
{   int a, b;
    scanf("%d,%d", &a, &b);
    Fun(a, b, Max);
    Fun(a, b, Min);
    Fun(a, b, Add);
    return 0;
}
void Fun(int x, int y, int (*f)(int, int))
{   int result;
    result = (*f)(x, y) ;
    printf("%d\n", result);
}
```

```
5,9↙
max=9
min=5
sum=14
```

```
int Max(int x, int y)
{
    printf("max=");
    return  x > y ? x : y;
}
```

```
int Min(int x, int y)
{
    printf("min=");
    return  x < y ? x : y;
}
```

```
int Add(int x, int y)
{
    printf("sum=");
    return  x + y;
}
```

函数指针的主要应用

- 函数指针的主要应用
  - 编写通用性更强的函数
- 典型实例1
  - 通用的计算任意函数定积分的函数
- 典型实例2
  - 通用的排序函数（既能升序，又能降序）