# 函数

**组成**：返回类型、函数名字、形参列表、函数体。 **调用**：函数名、调用运算符、实参列表。

## 函数组成

### 参数传递

函数会重新创建形参，用传入的实参对形参进行初始化。 参数初始化的机理和变量初始化是一样的。 实参被引用传递，函数被传引用调用。 实参被值传递，或者函数被传值调用。 常量形参和实参

1. 顶层const对于传值形参是没用的。比如void fcn(const int i)和void fcn(int i)就是重复的。 2 数组参数：以下三个语句效果是一样的。 void print(const int\*); void print(const int[]); void print(const int[10]); 数组参数指定数组长度：
2. 在最后一个字符后面跟一个空字符。
3. 传递指向数组首元素和尾后元素的指针。
4. 专门定义一个表示数组大小的形参。当函数不需要对数组元素执行写操作的时候，数组形参应该是指向const的指针。
5. 数组引用形参：形参绑定到数组上void f(int (&arr)[10]);。绑定了数组的大小。
6. 多维数组：传递的是指向数组首元素的指针，第二维的大小不能省略。 void print(int (\*matrix)[10], int rowSize); void print(int matrix[][10], int rowSize);

## main函数

1. 允许没有返回值。
2. 不能被调用。
3. 可以有两个参数argc和argv，分别表示参数数量和参数字符串数组。
4. 可以返回EXIT*SUCCESS或者EXIT*FAILURE。

## 递归函数

1. 递归循环

## 重载函数

## 内联函数

把较小规模的操作定义成函数而且又能避免函数调用的开销。

1. 函数可以确保行为的统一。
2. 如果有较好的函数命名，则函数比表达式更容易阅读和理解
3. 修改函数比逐一修改表达式方便
4. 函数可以被重复利用 ***注***：实际上实现起来是不是内联函数由编译器决定。

## constexpr函数

能用于常量表达式的函数。约定：

1. 函数返回值必须是字面值类型。
2. 函数中只有一条return语句。
3. 函数体内的语句在运行时不执行任何操作。
4. 实际上constexpr函数也是内联函数。
5. 通常定义在头文件中。

## assert预宏处理

assert是一种内联函数assert(expr);如果表达式为假，则中断程序执行，否则继续执行程序。

## NDEBUG预处理变量

如果定义了NDEBUG，assert什么也不做。#define NDEBUG `#ifdef NDEBUG // **func**是编译器定义的一个局部静态变量，用于存放函数的名字 cerr << **func** << “ : array size is ” << size << endl; //输出当前函数的名字。

# endif

除了\_\_func\_\_，还有\_\_FILE\_\_ \_\_LINE\_\_ \_\_TIME\_\_ \_\_DATE\_\_这几个预处理变量。if (word.size() <threshold) cerr << “Error: ” << **FILE** << “ : in function ” << **func** << “ at line” << **LINE** << endl << “ at “ << **TIME** << endl << “word ” << word << “is too short!” << endl; `

## 函数指针

指向函数而非对象，类型由函数的返回类型和形参类型共同决定，与函数名无关。 **声明**：returnType (\*pointerName)(parameter list); **赋值**：pointerName = functionName; **调用**：pointerName(param);, (\*pointerName)(param),functionName(param);三者等价。 **类型转换**：在定义函数时，不能定义函数类型的形参，但是可以定义函数指针的形参。如果有函数类型的实参被传入函数，那么函数类型会被转换成函数指针。 **类型推断**：decltype(functionName)能推断出函数类型，返回的是函数类型而不是函数指针。 typedef decltype(functionName) Func中的Func是函数类型 typedef decltype(functionName) \*FuncP中的FuncP是函数指针类型 **返回类型**：返回类型和函数形参一样，能用函数指针类型而不能用函数类型。 **尾置返回类型**：auto f1(int) -> int (*)(int*, int);