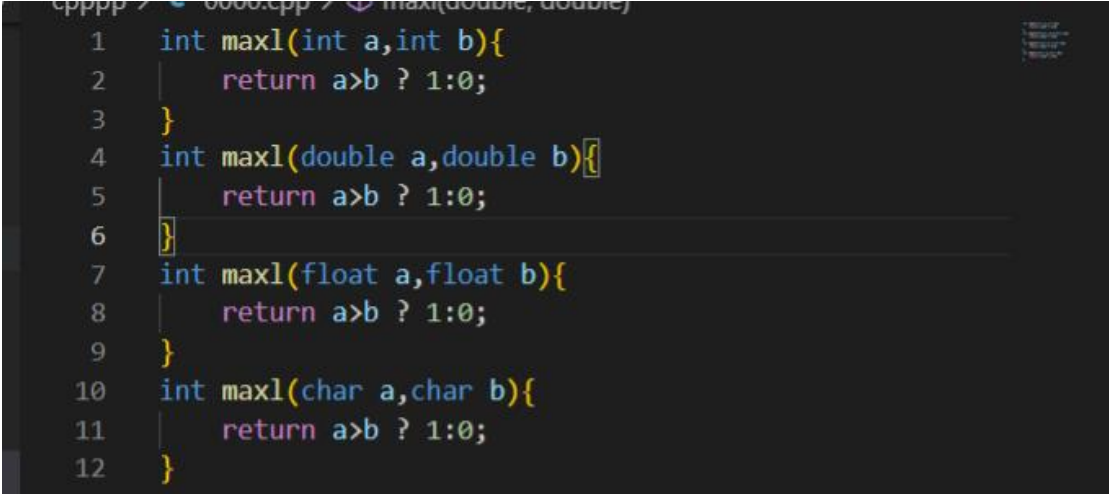


计算机学院 高级语言程序设计 课程实验报告

实验题目：c++简单程序设计（二）		学号：202200130048
日期：2023. 2. 28	班级： 6	姓名： 陈静雯
Email：1205037094@qq. com		
<p>实验目的：</p> <ol style="list-style-type: none">1. 练习 C++程序设计，熟悉 C++语法特点，数据表达。2. 掌握运用函数参数传递：值传递、地址传递（引用 与 指针的区别）。		
<p>实验软件和硬件环境：</p> <p>软件：Vscode</p> <p>硬件：11th Gen Intel (R) Core (TM) i7-1165G7 @ 2.80GHz 2.80 GHz</p> <p>64 位操作系统，基于 x64 的处理器</p>		
<p>实验步骤与内容：</p> <ol style="list-style-type: none">(1) 重载函数, 编写 4 个同名函数 maxl(2) 递归调用. 编写递归函数 fib(3) 观察递归调用的过程。 <p>在 Dev C++中调试过程：</p> <ol style="list-style-type: none">(a) 选择带 Debug 的编译选项：在工具栏右侧的编译器设置下拉框中选择，如：TDM-GCC 4.9.2 64-bit Debug(b) 设置/取消断点：在编辑窗口，移到某代码行上按 f4 或点击该行左端的行号。(c) 调试：按 F5，在左下的调试面板中点击相应的按钮进行程序调试操作。 <ol style="list-style-type: none">(4) 运行实验 2 素材中的程序		
<p>结论分析与体会：</p>  <pre>1 int maxl(int a,int b){ 2 return a>b ? 1:0; 3 } 4 int maxl(double a,double b){ 5 return a>b ? 1:0; 6 } 7 int maxl(float a,float b){ 8 return a>b ? 1:0; 9 } 10 int maxl(char a,char b){ 11 return a>b ? 1:0; 12 }</pre> <p>解释：四个同名函数，改变形参类型即可</p>		

```

3  long long fib(long long n)
4  {
5      if(n==1){
6          return 0;
7      }
8      else if(n==2){
9          return 1;
10     }
11     else if(a[n]){
12         return a[n];
13     }
14     else{
15         a[n]=fib(n-1)+fib(n-2);
16         return a[n];
17     }

```

解释：递归调用 斐波那契数列

```

问题  输出  调试控制台  终端
dsb5u13.lle' '--dbgExe=D:\mingw64\bin\gdb.exe' '--interpreter=mi
10111111 10000000 00000000 00000000
10111111 10000000 00000000 00000000
10111111 10000000 00000000 00000000
01000000 01001000 11110101 11000011
00111111 10000000 00000000 00000000
01000000 00011001 00011110 10111000 01010001 11101011 10000101 00011111
10111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000
11111111 11111111 11111111 11111111
01111111 11111111 11111111 11111111
10000000 00000000 00000000 00000000

```

解释：bits 函数先将&e 转为字符指针，再将其与 char (1) 即 00110001 从右往左不断移位补零后按位与结果为 true 输出 1，false 输出 0

```

sdebuglauncher.exe --stdin=Microsoft-MIEngine-In-0ugsviej.501
'--stdout=Microsoft-MIEngine-Out-4titdqv1.4ru' '--stderr=Micros
oft-MIEngine-Error-oaal415s.aws' '--pid=Microsoft-MIEngine-Pid-2
2sh1pdw.ppn' '--dbgExe=D:\mingw64\bin\gdb.exe' '--interpreter=mi
a=1 b=1
a=1 b=2

```

解释：第一个判断语句判断到 a==1 成立时不会判断下一个条件直接运行 if 里面的语句，所以 b 的值不变；第二个判断 (b=2) 先将 2 赋给 b，再进行按位或值不为零，故执行 if 下的语句 a=1, b=2

```

10
11 void swap1(int *a, int *b) {
12     int t = *a;
13     *a = *b;
14     *b = t;
15 }
16
17 int main() {
18     int x = 5, y = 10;
19     cout << "x = " << x << "    y = " << y << endl;
20     swap1(&x, &y);
21     cout << "x = " << x << "    y = " << y << endl;
22
23     return 0;
24
25 }

```

解释：用指针传递

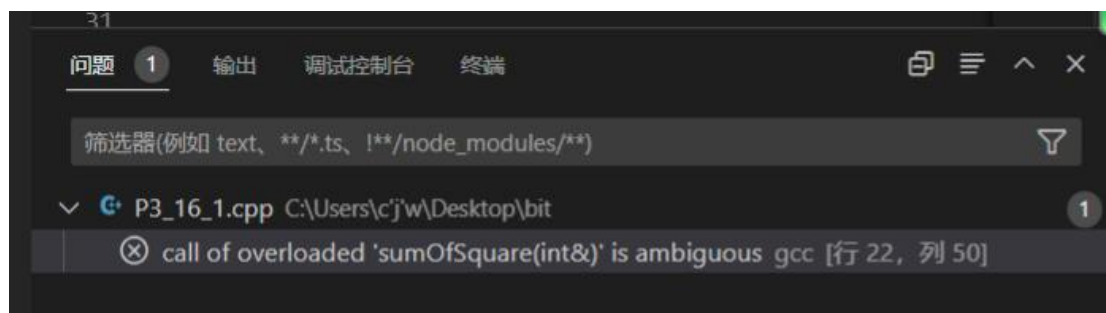
```

ons\ms-vscode.cpptools-1.14.3-win32-x64\debugAdapters\bin\window
sDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-5gnowxjz.1uv'
'--stdout=Microsoft-MIEngine-Out-lozz1oll.c1v' '--stderr=Microso
ft-MIEngine-Error-orhnnbeg.uyh' '--pid=Microsoft-MIEngine-Pid-p
yapureq.huu' '--dbgExe=D:\mingw64\bin\gdb.exe' '--interpreter=mi
,

The values are      7   12
The values are    107  112
The values are      7  112
PS D:\code_repository\codes

```

解释：in1 是 int 形参，只传递数值，in2 是引用变量，传递的就是这个变量本身，所以 in2 在函数内改变值就是 in2 本身值改变了



```

4
5 int sumOfSquare(int a, int b=1) {
6     return a * a + b * b;
7 }
8
9 int sumOfSquare(int a) {
10     return a * a ;
11 }

```

解释：重载函数 sumOfSquare 传入一个 m 对两个函数都适用，第一个就是相当于 (m, 1)，第二个就是 (m)，编译器无法判断所以报错

就实验过程中遇到的问题及解决处理方法，自拟 1—3 道问答题：

1.

```
2  template<typename T>
3  void Bits(T const& e)
4  {
5      int n(sizeof(T));
6      char* ch=(char*)&e;
7      for(int i(n-1),j;i>=0;--i)
8      {
9          for(j=7;j>=0;--j)
10             ch[i]&(char(1)<<j) ? std::cout<<1 : std::cout<<0;
11             std::cout<<' ';
12         }
13         std::cout<<"\n";
14     }
```

问：(char*) &e 赋给 char*ch 后，ch[i]具体存的是什么？

答：应该是 e 的地址，int 有四个字节，ch 就从 3 到 0，会输出 4 组

问：与 char (1) 按位与是将 1 不断左移还是将 00110001 (1 的 ASCII 码) 左移补位再按位与？

答：应该是将 int 1 转为 char，就是字符 1，即 00110001