计算机学院 高级程序语言设计 课程实验报告

实验题目：C++简单程序设计（二）		学号：202300130150
日期：2024-3-14	班级： 4	姓名： 王成意

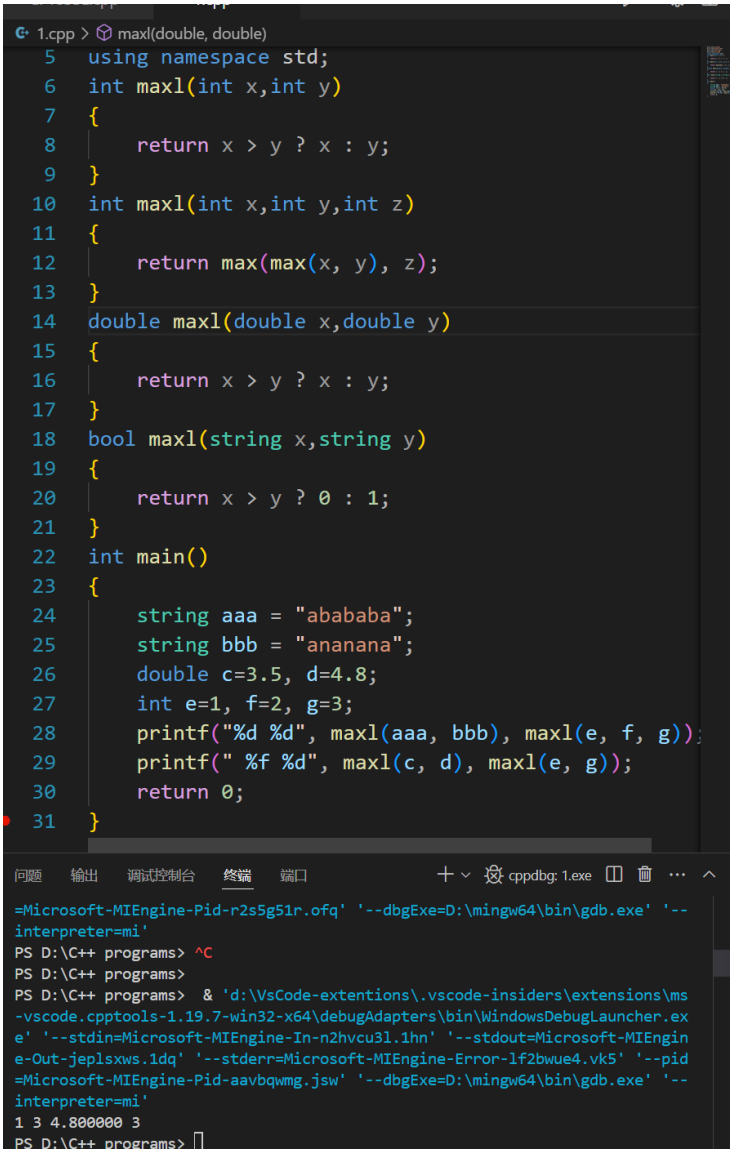
实验目的：

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）。

实验步骤与内容：

实验步骤：

（2）重载函数，编写 4 个同名函数 maxl



```
1.cpp > maxl(double, double)
 5  using namespace std;
 6  int maxl(int x,int y)
 7  {
 8      return x > y ? x : y;
 9  }
10  int maxl(int x,int y,int z)
11  {
12      return max(max(x, y), z);
13  }
14  double maxl(double x,double y)
15  {
16      return x > y ? x : y;
17  }
18  bool maxl(string x,string y)
19  {
20      return x > y ? 0 : 1;
21  }
22  int main()
23  {
24      string aaa = "abababa";
25      string bbb = "ananana";
26      double c=3.5, d=4.8;
27      int e=1, f=2, g=3;
28      printf("%d %d", maxl(aaa, bbb), maxl(e, f, g));
29      printf(" %f %d", maxl(c, d), maxl(e, g));
30      return 0;
31  }
```

```
问题  输出  调试控制台  终端  端口   cppdbg: 1.exe
=Microsoft-MIEngine-Pid-r2s5g51r.ofq' '--dbgExe=D:\mingw64\bin\gdb.exe' '--
interpreter=mi'
PS D:\C++ programs> ^C
PS D:\C++ programs>
PS D:\C++ programs>  & 'd:\VsCode-extentions\.vscode-insiders\extensions\ms
-vscode.cpptools-1.19.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.ex
e' '--stdin=Microsoft-MIEngine-In-n2hvcu3l.1hn' '--stdout=Microsoft-MIEngin
e-Out-jeplsxws.1dq' '--stderr=Microsoft-MIEngine-Error-lf2bwue4.vk5' '--pid
=Microsoft-MIEngine-Pid-aavbqwmg.jsw' '--dbgExe=D:\mingw64\bin\gdb.exe' '--
interpreter=mi'
1 3 4.800000 3
PS D:\C++ programs>
```

（4）递归调用，编写递归函数 fib

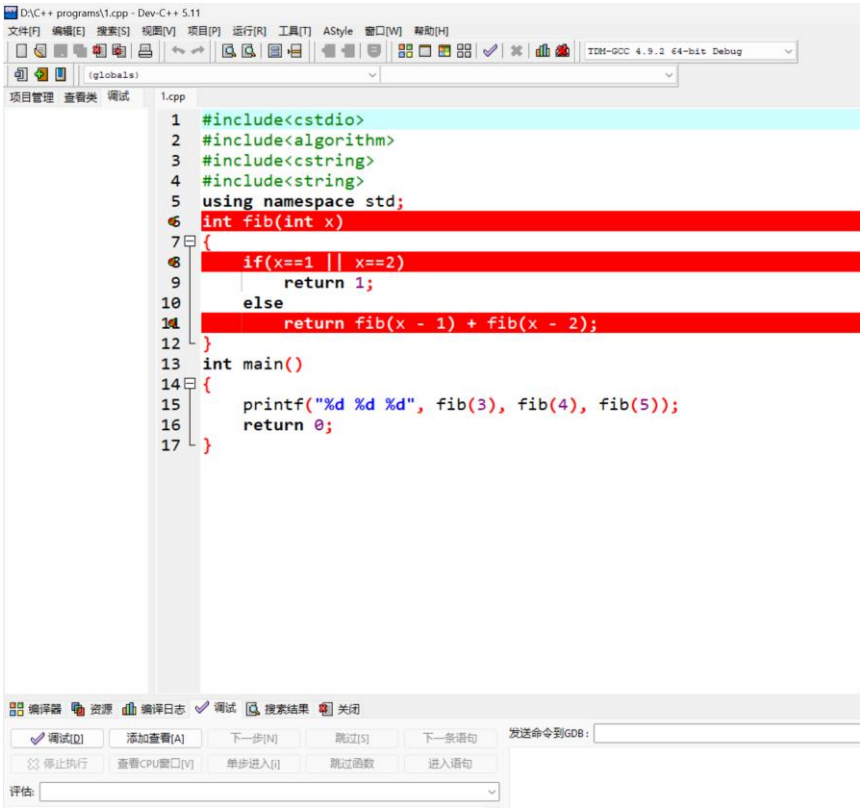
1.cpp > main()

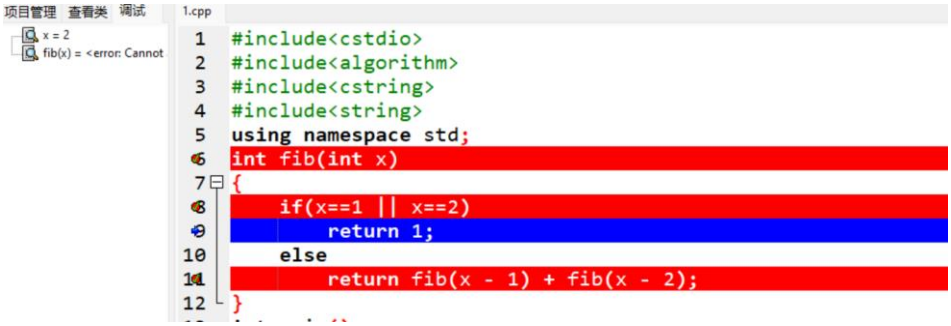
```cpp
#include<cstdio>
#include<algorithm>
#include<cstring>
#include<string>
using namespace std;
int fib(int x)
{
    if(x==1 || x==2)
        return 1;
    else
        return fib(x - 1) + fib(x - 2);
}
int main()
{
    printf("%d %d %d", fib(3), fib(4), fib(5));
    return 0;
}
```

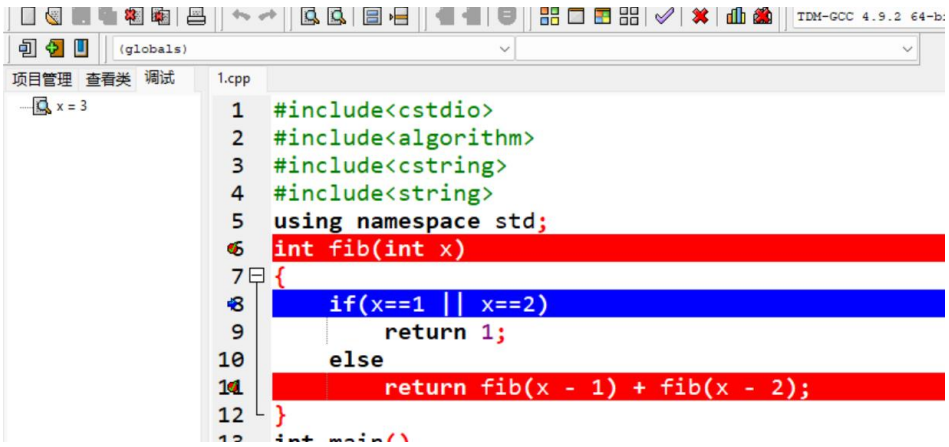
问题 输出 调试控制台 终端 端口 cppdbg: 1.exe

```
interpreter=mi'
1 3 4.800000 3
PS D:\C++ programs> ^C
PS D:\C++ programs>
PS D:\C++ programs>  & 'd:\VsCode-extentions\.vscode-insiders\extensions\ms-vscode.cpptools-1.19.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-z0l3u0d4.l0o' '--stdout=Microsoft-MIEngine-Out-dyzfmdmy.ag0' '--stderr=Microsoft-MIEngine-Error-gxtxnkod.fz4' '--pid=Microsoft-MIEngine-Pid-zpduox1v.j35' '--dbgExe=D:\mingw64\bin\gdb.exe' '--interpreter=mi'
2 3 5
PS D:\C++ programs>
```

（5）观察递归调用的过程。





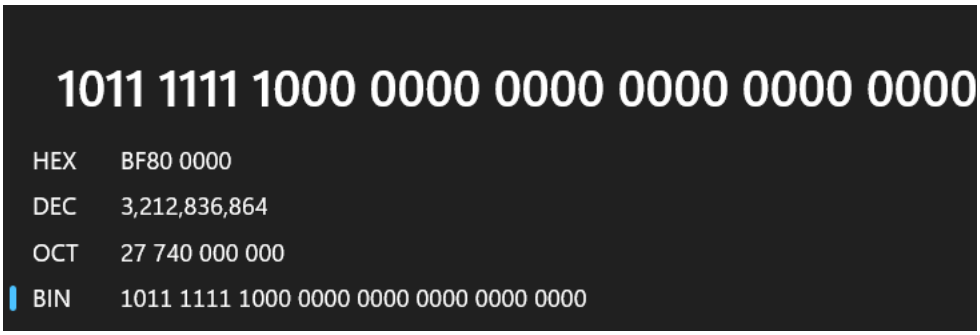


5->4->3->2(return 1)->1(return 1)->2(return 1)->3->2(return 1)->1(return 1)->return 5

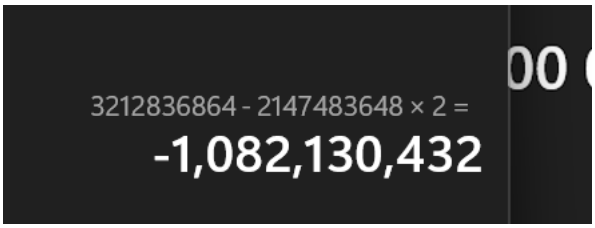
运行实验 2 素材中的程序：
同一个二进制数，不同类型的解释，浮点数表达；
bit.cpp 解释运行结果

```
template<typename T>
void Bits(T const& e)
{
    int n(sizeof(T));//取字节数
    char* ch=(char*)&e;//取地址
    for(int i(n-1),j;i>=0;--i)//字节枚举
    {
        for(j=7;j>=0;--j)//byte枚举，判断01
            ch[i]&(char(1)<<j) ? std::cout<<1 : st
        std::cout<<' ';//每个字节断开
    }
    std::cout<<"\n";//每组数据断开
}
```

所以对于 u（unsigned），01 串表示其二进制数



W 因为加上了符号位，所以比 u 小 2^32



对于 f，因为它是浮点数，编码方式和 int 不一样，第一位是符号位：

```
10111111 10000000 00000000 00000000
```

后面八位是指数位

```
10111111 10000000 00000000 00000000
```

偏移量是-127，表示 0，也就是 2^0=1；
尾数位全是 0，所以 ans=-1*2^0+0=-1.0.
接下来是 a（3.14）和 b（1.0）

```
01000000 01001000 11110101 11000011
00111111 10000000 00000000 00000000
```

再往下是 d，e（double），所以有 8 个字节：

```
01000000 00011001 00011110 10111000 01010001 11101011 10000101 00011111
10111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000
```

最后，x 由于存在符号位，编码是 0xffffffff;z 是 0x7fffffff;
Z+1 编码按照二进制加法，溢出。

```
11111111 11111111 11111111 11111111
01111111 11111111 11111111 11111111
10000000 00000000 00000000 00000000
```

short_circuit.cpp　解释运行结果
第一个 if 输出满足第一个，结束判断（不执行（(b=2)==2)），按照‘or‘输出；
第二个 if 输出括号里进行了或运算，1|1=1，即 if（1），满足条件；在计算过程中执行了(b=2)==2 语句，b 变成 2，输出。

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

int main()
{
    int a=1,b=1;
    if( a==1 || (b=2)==2)
        cout<<"a="<<a<<" b="<<b<<endl;

    if( a==1 | (b=2)==2)
        cout<<"a="<<a<<" b="<<b<<endl;

    return 0;
}
```

```
问题  输出  调试控制台  终端  端口   cppdbg: short_circuit.exe

2a=1 b=2
PS D:\C++ programs> ^C
PS D:\C++ programs>
PS D:\C++ programs>  & 'd:\VsCode-extentions\.vscode-insiders\extensions\
-vscode.cpptools-1.19.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.
e' '--stdin=Microsoft-MIEngine-In-fyvlvqrz.x4n' '--stdout=Microsoft-MIEng
e-Out-3q1onps2.fsp' '--stderr=Microsoft-MIEngine-Error-laa3nd1o.bsc' '--p
=Microsoft-MIEngine-Pid-gey2edmh.04x' '--dbgExe=D:\mingw64\bin\gdb.exe' '
interpreter=mi'
a=1 b=1
a=1 b=2
```

p3_12. cpp 请增加一个 swap 函数，形参用指针传递实现数据交换。

```cpp
//3_12.cpp
#include <iostream>
using namespace std;

void swap(int &a, int &b) {
    int t = a;
    a = b;  //使用引用，如同使用变量
    b = t;
}
void swap2(int *a,int *b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;
}
int main() {
    int x = 5, y = 10;
    cout << "x = " << x << "    y = " << y << endl;
    swap2(&x, &y);
    cout << "x = " << x << "    y = " << y << endl;

    return 0;

}

```

```
问题  输出  调试控制台  终端  端口                cppdbg: p3_12.exe
x = 10    y = 5
PS D:\C++ programs> ^C
PS D:\C++ programs>
PS D:\C++ programs>  & 'd:\VsCode-extentions\.vscode-insiders\extensions\ms
-vscode.cpptools-1.19.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.ex
e' '--stdin=Microsoft-MIEngine-In-qcx4tq3a.emf' '--stdout=Microsoft-MIEngin
e-Out-z3c11m5h.csh' '--stderr=Microsoft-MIEngine-Error-vlx5ke54.otf' '--pid
=Microsoft-MIEngine-Pid-hdsarasi.apl' '--dbgExe=D:\mingw64\bin\gdb.exe' '--
interpreter=mi'
x = 5    y = 10
x = 10    y = 5
```

p3_13. cpp 解释运行结果

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

void fiddle(int in1, int &in2) {
    in1 = in1 + 100;
    in2 = in2 + 100;
    cout << "The values are ";
    cout << setw(5) << in1;
    cout << setw(5) << in2 << endl;
}

int main() {
    int v1 = 7, v2 = 12;
    cout << "The values are ";
    cout << setw(5) << v1;
    cout << setw(5) << v2 << endl;
    fiddle(v1, v2);
    cout << "The values are ";
    cout << setw(5) << v1;
    cout << setw(5) << v2 << endl;
    return 0;
}
```

```
问题  输出  调试控制台  终端  端口   cppdbg: p3_13.exe

PS D:\C++ programs>  & 'd:\VsCode-extentions\.vscode-insiders\extensions\
-vscode.cpptools-1.19.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.
e' '--stdin=Microsoft-MIEngine-In-hzwunhgx.1p0' '--stdout=Microsoft-MIEng
e-Out-krscapb1.5me' '--stderr=Microsoft-MIEngine-Error-ocua25k5.kfv' '--p
=Microsoft-MIEngine-Pid-vdiznajy.st3' '--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:\C++ programs>
```

In1 是形参，函数里改变的值不会影响其真实值；in2 在函数里传的实参，函数里的改变会影响其真实值。

p3_16_1.cpp 为何编译出错？

```
正在启动生成...
cmd /c chcp 65001>nul && D:\mingw64\bin\g++.exe -fdiagnostics-color=always
-g "D:\C++ programs\tmp\P3_16_1.cpp" -o "D:\C++ programs\tmp\P3_16_1.exe" -
std=c++17
D:\C++ programs\tmp\P3_16_1.cpp: In function 'int main()':
D:\C++ programs\tmp\P3_16_1.cpp:22:50: error: call of overloaded 'sumOfSqua
re(int&)' is ambiguous
  cout << "Their sum of square: " << sumOfSquare(m) << endl;
                                                 ^
D:\C++ programs\tmp\P3_16_1.cpp:5:5: note: candidate: 'int sumOfSquare(int,
 int)'
 int sumOfSquare(int a, int b=1) {
     ^~~~~~~~~~~
D:\C++ programs\tmp\P3_16_1.cpp:9:5: note: candidate: 'int sumOfSquare(int)
'
 int sumOfSquare(int a) {
     ^~~~~~~~~~~
```

call of overloaded 'sumOfSquare(int&)' is ambiguous:

```
<< sumOfSquare(m, n) <<
<< sumOfSquare(m) << en
```

```
int sumOfSquare(int a, int b=1) {
    return a * a + b * b;
}

int sumOfSquare(int a) {
    return a * a ;
}
```

就是说，只带一个 m 是模糊的：不知道是用上面的函数还是下面的函数。

验证：删除第二个函数正常运行：

```cpp
//3_16.cpp
#include <iostream>
using namespace std;

int sumOfSquare(int a, int b=1) {
    return a * a + b * b;
}

double sumOfSquare(double a, double b) {
    return a * a + b * b;
}

int main() {
    int m, n;
    cout << "Enter two integer: ";
    cin >> m >> n;
    cout << "Their sum of square: " << sumOfSquare
    cout << "Their sum of square: " << sumOfSquare

    double x, y;
    cout << "Enter two real number: ";
    cin >> x >> y;
    cout << "Their sum of square: " << sumOfSquare

    return 0;
}
```

```
题  输出  调试控制台  终端  端口   cppdbg: P3_16_1.exe
S D:\C++ programs>  & 'd:\VsCode-extentions\.vscode-insiders\extensions\ms
vscode.cpptools-1.19.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.ex
' '--stdin=Microsoft-MIEngine-In-uvlec4xx.3nj' '--stdout=Microsoft-MIEngin
-Out-24bal4uo.nia' '--stderr=Microsoft-MIEngine-Error-rqoygqt2.5qu' '--pid
Microsoft-MIEngine-Pid-b1gyv0lb.cky' '--dbgExe=D:\mingw64\bin\gdb.exe' '--
nterpreter=mi'
nter two integer: 6 6
heir sum of square: 72
heir sum of square: 37
nter two real number: 3.0 3.0
heir sum of square: 18
S D:\C++ programs>
```

结论分析与体会：
通过实验我收获了很多，包括如何获得机器码，如何调试等等。