实验二 C++对 C 的扩充

实验人：罗啸
实验地点：电气学院 412
指导老师：邝先验
学号：2420173095

2.1 实验目的

1．进一步了解和熟悉 VC++6.0 开发环境，学会在 VC++6.0 环境下调试程序；
2．熟悉 C++中简单的标准输入输出函数的实用；
3．理解 const 修饰符的作用，并学会应用 const 修饰符；
4．理解内置（内联）函数的优缺点并学会使用内置函数；
5．理解和使用函数重载以及带默认参数的函数；
6．使用 new 和 delete 进行动态内存管理；
7．理解和使用引用。

2.2 实验内容

2.2.1 程序阅读

1．理解下面的程序，并在 VC++6.0 下运行查看结果，回答程序后面的问题。

```
#include <iostream >
using namespace std;
int max_def(int x, int y)
{
    return (x>y?x:y);
}
int max_def(int x, int y, int z)
{
    int temp = 0;
    return (temp=(x>y?x:y))>z?temp:z;
}
double max_def(double x, double y)
{
    return (x>y?x:y);
}
int main()
{
    int x1 = 0;
    int x2 = 0;
    double d1 = 0.0;
```

```
    double d2 = 0.0;
    x1 = max_def(5,6);
    x2 = max_def(2,3,4);
    d1 = max_def(2.1,5.6);
    d2 = max_def(12.3,3.4,7.8);----------------------------------------------------①
    cout<<"x1="<<x1<<endl;
    cout<<"x2="<<x2<<endl;
    cout<<"d1="<<d1<<endl;
    cout<<"d2="<<d2<<endl;----------------------------------------------------②
    return 1;
}
```

问题一：上述程序的输出结果是什么？

答：

x1=6

x2=4

d1=5.6

d2=12

问题二：哪些情况可以参与函数的重载？

答：

参数类型改变，参数个数改变

问题三：①处调用的是哪个函数？

答：

double max_def(double x, double y)；

问题四：②处的输出结果为什么是 d2=12，而不是 d2=12.3？

答：

因为调用的是 int 类型的函数 int max_def(int x, int y, int z)，返回值为 int 型

2．理解下面的程序，并在 VC++6.0 下运行查看结果，回答程序后面的问题。

```
#include <iostream>
using namespace std;
int main()
{
    int *p1 = new int; ----------------------------------------------------①
    int *p2 = new int(0); ----------------------------------------------------②
    char *p3 = new char[64]; ----------------------------------------------------③
return 1;
}
```

问题一：①、②、③处动态申请内存分别代表什么不一样的意思？

答：

1.申请了一个 int 类型的空间，，指针 p1 指向它的位置；

2.申请了一个 int 类型的空间并初始化它，值为 0，指针 p2 指向它的位置；

3.申请了一块 char 型数组空间，长度为 64，p3 指向该数组首地址

问题二：该程序存在什么隐患？改正该程序使隐患消除。

答：

动态内存未清除，可以在程序结尾 delete 它们。

```
delete p1;
delete p2;
delete[] p3;
```

3．理解下面的程序，并在 VC++6.0 下运行查看结果，回答程序后面的问题。

```
#include <iostream>
using namespace std;
void swap(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main()
{
    int i = 5;
    int j = 10;
    cout<<"Before swap: i="<<i<<",j="<<j<<endl;
    swap(i,j); ----------------------------------------------------①
    cout<<"After the first swap: i="<<i<<",j="<<j<<endl;
    swap(&i,&j); ----------------------------------------------------②
```

```
    cout<<"After the second swap: i="<<i<<",j="<<j<<endl;
    return 1;
}
```

问题一：上述程序的输出结果是什么？

答：Before swap: i=5,j=10

After the first swap: i=5,j=10

After the second swap: i=10 ,j=5

问题二：①处函数调用后并不能实现两个数的交换，而②处却可以，为什么？

答：

因为1处只是对函数里的a,b进行了赋值，在函数里面进行了交换，而对函数外面的a,b无影响；2处利用指针，传入参数的地址，对参数进行了改变，故进行了交换。

问题三：②处调用的是哪个重载函数？

答：调用的是 void swap(int *a, int *b)；

2.2.2 程序设计

1．使用函数重载的方法定义两个重名函数，分别求出整形数平面间两点间距离和双精度平面间两点间距离，如果没有输入第二点的坐标则默认为圆点（0,0）。

```
#include <iostream>
using namespace std;
double length(double a,double b, double c, double d)   //双精度，输入两个坐标
{
    double len;
    len = sqrt((c - a)*(c - a) + (d - b)*(d - b));
    return len;
}
int length(int a, int b, int c, int d)                  //整型，输入两个坐标
{
    int len;
    len = sqrt((c - a)*(c - a) + (d - b)*(d - b));
    return len;
}
double length(double a, double b)                       //双精度，输入一个坐标
{
    double len;
    len = sqrt(a*a + b * b);
```

```
    return len;
}
int length(int a, int b)                                    //整型，输入一个坐标
{
    int len;
    len = sqrt(a*a + b * b);
    return len;
}
int main()
{
    int a, b, c, d;
    cin >> a >> b >> c >> d;
    length(a, b, c, d);
    return 1;
}
```

2．设计一个函数：exchange(float x, float y, float z)，当调用 exchange(a,b,c)时，将 a 的内容赋值给 b，b 的内容赋值给 c，c 的内容赋值给 a，要求采用引用的方式来实现。

```
#include <iostream>
using namespace std;
float exchange(float &a, float &b, float &c)
{   float temp;
    temp = a; a = c; c = b; b = temp;
    return 0;
}
int main()
{
    float a, b, c;
    cin >> a >> b >> c;
    exchange(a, b, c);
    cout << a << endl;
    cout << b << endl;
    cout << c << endl;
    return 1;
}
```

2.3 思考题

1．自己设计一个程序，测试 const 的三种用法：指向常量的指针，常指针，指向常量的常指针。

```
#include<iostream>
using namespace std;
int main()
{
    int a = 2;
    const int *p=int(0);                    //指向常量的指针，地址可变，值不可变
    int* const p1=&a;                       //常指针，地址不可变，值可变
    const int const*p2;                  //指向常量的常指针，地址和数值不可变
    return 0;
}
```

2．编写一个函数，实现两个字符串变量的交换，要求参数用引用。

```
#include <iostream>
using namespace std;
void swap(char &ch1, char &ch2)
{
    char temp;
    temp = ch1; ch1 = ch2; ch2 = temp;
}
int main()
{
    //限制字符串长度为 100，进行交换
    char ch1[100], ch2[100];
    cin >> ch1;
    cin >> ch2;
    for (int i = 0; i < 100; i++)
        swap(ch1[i], ch2[i]);
    cout << ch1 << endl;
    cout << ch2 << endl;
    return 1;
}
```