

C++作业

罗悦 2016220304022

习题 2.5

答：

*p 表达的含义是 p 这个指针所指向的地址表示的值，即 *p 表示 a 的值，即 *p=1。

&a 表示的意思是 &a 所指向的地址所表示的值，即 a 的值，即 *&a=1。

&*p 表示 *p 的地址，就是指 p 这个指针的内容所表示的地址。

++*p 表示先取出 p 指针所指向的地址的内容值，并加 1。

*p-- 表示先将 p 指针内容减一形成一个新地址，再取出新地址所表示的值。

(*p)-- 表示先取出 p 指针所指向的地址的内容值，并减 1。

习题 2.7

答：所编写程序如下：

```
#include <iostream>
using namespace std;
int main()
{
    int a[10];
    int i;
    for (i=0;i<10;i++)
    {
        cin >> a[i]; // 键盘输入 10 个数
    }
    int *p=&a;        //定义一个指针访问数组
    for(i=0;i<10;i++,p++)
```

```

    {
        printf("%d\n", *p);
    }
    int j, n, k, m;
    n=10;
    for(m=0; m<n-1; m++) {        //冒泡算法进行排序
        for(j=0; j<n-1; j++) {
            if(a[j]>a[j+1]) {
                k=a[j+1];
                a[j+1]=a[j];
                a[j]=a[j+1];
            }
        }
    }
    return 0;
}

```

习题 2.10

答:

$a=b=a$ 即 $a=(b=a)$ 即先把 a 的值赋给 b , 即 $b=2$, 然后将 b 把值赋给 a , 即 $a=2$ 。

$b-=a++$ 即 $b=b-(a++)$ 即先对 b 进行赋值操作, 即 $b=4-2=2$, 然后然对 a 的值进行加一操作, 即 $a=2+1=3$ 。

$a-=b*=b/=2$ 即先进行 $b/=2$ 操作, 即 $b=b/2$ 即 $b=4/2=2$, 然后进行 $b*=2$ 操作, 即 $b=b*2$ 即 $b=2*2=4$, 然后进行 $a-=4$ 操作, 即 $a=a-4$ 即 $a=2-4=-2$ 。

$b>=a$ 这是一个布尔表达式, 即判断 b 是否大于等于 a , 4 大于 2, 所以这个布尔表达式结果为 `true`。

$a\&\&a>0$ 这是一个与操作, 即判断 $\&\&$ 左右两边的表达式的结果是否都为 `true`, 这里 a 不等于 0 且 $a>0$, 所以这个判断结果为 `true`。

$a=b/10? 1: 0$ 因为 b 为整型, 所以 $b/10$ 为 0, 所以结果为 `false`,

所以将 “:” 右边的值赋给 a，即 a=0。

b=a/b，a*b 即先计算 a/b 的值赋给 b，然后再计算 a*b 的值赋给 b，
最终 b=4。

习题 3.2

答：

我设计了一个 2X4 和 4X4 的的矩阵相乘的算法：

```
#include<iostream>
using namespace std;
#define SIZE_M 2
#define SIZE_S 4

int main() {

    int matrix_a[SIZE_M][SIZE_S]={ {1, 2, 3, 4}, {4, 5, 6, 7}};    //a 矩阵 2X4
    int matrix_b[SIZE_S]
[SIZE_S]={ {1, 2, 3, 4}, {5, 6, 7, 8}, {1, 2, 3, 4}, {5, 6, 7, 8}};    //b 矩阵 4X4
    int matrix_result[SIZE_M][SIZE_S];    //结果矩阵 2X4

    for(int m=0;m<SIZE_M;m++) {
        for(int s=0;s<SIZE_S;s++) {
            matrix_result[m][s]=0;    //初始化
            for(int n=0;n<SIZE_S;n++) {
                matrix_result[m][s]+=matrix_a[m][n]*matrix_b[n][s];
            }
        }
    }
    for(int m=0;m<SIZE_M;m++) {
        for(int s=0;s<SIZE_S;s++) {
            cout<<matrix_result[m][s]<<"\t";
        }
        cout<<endl;
    }
    return 0;
}
```

习题 3.7

答:

设计的简单计算器如下算法:

```
#include <iostream>
using namespace std;
int main()
{
    double a,b,r;
    char c;
    cin >> a;
    if(!cin) error("no firs operand");
    while (cin >> c)
    {
        if (c != ';'')
        {
            cin >> r;
        }
        if (!cin)
        {
            error("no second operand");
        }
    }
    switch(c) {
        case '+': r = a + b; break;
        case '-': r = a - b; break;
        case '*': r = a * b; break;
        case '/': r = a / b; break;
        case '%': r = a % b; break;
        default: r = 0;
    }

    cout<<r<<endl;
    return 0;
}
```

习题 4.2

答:

最终: a=2, b=4, c=4。

习题 4.3

答：

设计程序算法如下所示：

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

int *p;
void SomeFunction1() {    //动态内存分配函数
    p = new int;
    *p = 17;
}
void SomeFunction2() {    //动态内存释放函数
    delete p;
}

int add1(int *p1,int *p2){ //用指针实现参数和返回值的传递
    int a=*p1;
    int b=*p2;
    int c=a+b;
    return c;
}
int& add2(int &a,int &b){ //用引用实现参数和返回值的传递
    int c=a+b;
    return c;
}

int main()
{
    SomeFunction1();    //在 main() 中实现分配的动态内存
    cout<<*p<<endl;
    SomeFunction2();
    int a=1;
    int b=2;
    int *p1=&a;
    int *p2=&b;
    cout<<add1(p1,p2)<<endl;
    cout<<add2(a,b)<<endl;
    return 0;
}
```

习题 4.4

答：所编写程序如下所示：

```
#include <iostream>
using namespace std;
int compare(int a,int b)    //整型
{
    int c;
    if(a>b) {
        c=1;
    }
    else if(a==b) {
        c=0;
    }
    else{
        c=-1;
    }
    return c;
}

double compare(double a,double b)    //双精度
{
    int c;
    if(a>b) {
        c=1;
    }
    else if(a==b) {
        c=0;
    }
    else{
        c=-1;
    }
    return c;
}

float compare(float a,float b)    //浮点数
{
    int c;
    if(a>b) {
        c=1;
    }
    else if(a==b) {
        c=0;
    }
    else{
```

```

        c=-1;
    }
    return c;
}

int compare(int *a,int *b)  //指针类型
{
    int d=*a;
    int e=*b;
    int c;
    if(d>e){
        c=1;
    }
    else if(d==e){
        c=0;
    }
    else{
        c=-1;
    }
    return c;
}

int main()
{
    int c;
    int a=1;
    int b=2;
    c=compare(1,2);           //对重载函数进行测试
    cout<<c<<endl;
    c=compare(1.2,1.1);
    cout<<c<<endl;
    c=compare(3,3);
    cout<<c<<endl;
    int *p1=&a;
    int *p2=&b;
    c=compare(p1,p2);
    cout<<c<<endl;

    return 0;
}

```

