

上机测验一

1. 编写C或C++程序，计算从键盘输入的一系列正整数的和，并输出。要求先输入整数的个数（所涉及的数据均在int型数据范围以内）。

```
int main()
{
    int m;
    cin >> m;                //scanf("%d", &m);
    int sum=0;
    for(int i=0; i < m; ++i)
    {
        int n;
        cin >> n;            //scanf("%d", &n);
        sum += n;
    }
    cout << sum << endl;     //printf("%d \n", sum);
    return 0;
}
```

-
2. 小强每 3 天爬一次紫金山，小明每 7 天爬一次紫金山，每次他们都会到头陀铃歇一会儿，假定2019年，小强在第 `xdate` 天开启了爬山运动，小明在第 `mdate` 天开启了爬山运动，编写C或C++程序判断他俩今年在头陀铃会不会相遇。输入 `xdate` 和 `mdate` 两个小于 366 的整数，输出 `y` 或 `n` 表示会或不会相遇。

```
int main()
{
    int xdate, mdate;
    cin >> xdate >> mdate;    //scanf("%d%d", &xdate, &mdate);
    for(int i=xdate; i < 366; i += 3)
        for(int j=mdate; j <= i; j += 7)
            if(i == j)
            {
                cout << "y" << endl;    //printf("y\n");
                return 0;
            }
    cout << "n" << endl;    //printf("n\n");
    return 0;
}
```

```
int main()
{
    int xdate, mdate;
    cin >> xdate >> mdate;
    for(int i=xdate; i < 366; i += 3)
        for(int j=mdate; j <= i; j += 7)
            if(i == j)
            {
                cout << "y" << endl;
                goto END;
            }
    cout << "n" << endl;
    END: return 0;
}
```

```
int main()
{
    int xdate, mdate, flag=0;
    cin >> xdate >> mdate;
    for(int i=xdate; i < 366; i += 3)
        for(int j=mdate; j <= i; j += 7)
            if(i == j)
                flag = 1;
    if(flag == 1)
        cout << "y" << endl;
    else
        cout << "n" << endl;
    return 0;
}
```

错误的做法

```
int main()
{
    int xdate, mdate;
    cin >> xdate >> mdate;
    for(int i=xdate; i < 366; i += 3)
        for(int j=mdate; j <= i; j += 7)
            if(i == j)
            {
                cout << "y" << endl;
                break; // 只能退出内层循环
            }
    cout << "n" << endl;
    return 0;
}
```

错误的做法

```
int main()
{
    int xdate, mdate;
    cin >> xdate >> mdate;
    for(int i=xdate; i < 366; i += 3)
    {
        for(int j=mdate; j <= i; j += 7)
            if(i == j)
            {
                cout << "y" << endl;
                break;
            }
        if(i == j)break; //退出外层循环后会继续输出 n
    }
    cout << "n" << endl;
    return 0;
```

错误的做法

```
int main()
{
    int xdate, mdate;
    cin >> xdate >> mdate;
    for(int i=xdate; i < 366; i += 3)
        for (int j=mdate; j <= i; j += 7)
            if (i == j)
                cout << "y" << endl;
            else
                cout << "n" << endl;
    return 0;
}
```

//过早下结论

循环的优化

- 循环操作常常比较耗时，所以编译器往往会做一些优化，以便提高程序的运行效率。例如，

```
for (i = 0; i < 10000; ++i)  
    s = a+b;
```

上面这种重复计算会被编译器直接优化成：

```
s = a+b;
```

- 如果希望实现延时，可以关闭编译器的优化功能（项目-属性-优化-Disabled）

提取与循环无关的计算

- 程序员也可以有意识地优化循环程序，以便提高程序的运行效率。例如，

```
int s=0, m, n;  
scanf("%d%d", &m, &n);  
for(int i = 0; i < m*n+100; ++i)  
    s += i;
```

可以优化成：

```
int s=0, m, n;  
scanf("%d%d", &m, &n);  
int l = m*n;  
for(int i = 0; i < l+100; ++i)  
    s += i;
```

嵌套循环应遵循“外小内大”原则

- 程序员自己也可以进行适当的优化，以便提高程序的运行效率。例如，

```
for(int i = 0; i < 1000; ++i)
    for(int j = 0; i < 10; ++j)
        s += i*j;
```

可以优化成：

```
for(int j = 0; j < 10; ++j)
    for(int i = 0; i < 1000; ++i)
        s += i*j;
```

Thanks!

