

## 中国矿业大学计算机学院实验报告

|   |                             |      |          |
|---|-----------------------------|------|----------|
| 课程名称  | 高级语言程序设计                    |      |          |
| 实验名称  | 高级语言程序设计实验(11月12日5-8节)-薛猛老师 |      |          |
| 班级  | 信息安全19-01班                  | 姓名   | 许万鹏      |
|   |                             | 学号   | 05191643 |
| 仪器组号  | 66                          | 实验日期 | 11月12日   |
| 实验报告要求：1. 实验目的  |                             |      |          |
| 2. 实验内容（题目描述，源代码，运行截图，调试情况）   |                             |      |          |
| 3. 实验体会   |                             |      |          |
| 一、实验目的  |                             |      |          |
| 1) 掌握函数的定义、声明和调用方法；   |                             |      |          |
| 2) 掌握参数传递机制、结果返回的方法；  |                             |      |          |
| 3) 熟悉变量的存储类型及有效范围；  |                             |      |          |
| 4) 掌握局部普通变量、静态全局变量和全局变量的用法。   |                             |      |          |
| 二、实验内容  |                             |      |          |
| 1、第一题   |                             |      |          |
| 1.1 题目描述  |                             |      |          |
| 编写具有如下原型的函数：bool f(long x);其功能为：若整数 x 仅由偶数字(0、2、4、6、8)组成时(如 x=26480)，函数返回 true，否则返回 false(如当 x=22034 时)。并编制主函数对它进行调用。 |                             |      |          |
| 1.2 源代码   |                             |      |          |
| <pre>#include &lt;iostream&gt;  using namespace std;  bool f(long x);  int main()  {      long n;</pre>               |                             |      |          |

```

        cin>>n;

        cout<<boolalpha<<f(n);

        return 0;
    }

bool f(long x)
{
    int num;

    bool flag=true;

    while(x)
    {

        num=x%10;

        if(num%2!=0) flag=false;

        x/=10;

    }

    return flag;
}

```

### 1.3 运行截图

问题    输出    调试控制台    终端

```

> Executing task: D:\Codefield\CODE_Cpp\Cpp_Single\No.3\bin\1.exe <

26480
true
按任意键关闭终端。

```

### 1.4 调试情况

Accepted, 见上图。

## 2、第二题

### 2.1 题目描述

编写具有如下原型的函数：void find(int i, int n); 实现从 i 开始找起，连续找出 n 个素数并显示在屏幕上。如：实参为 10 和 8 后，输出的 8 个素数应为：11, 13, 17, 19, 23, 29, 31, 37

### 2.2 源代码

```
#include <iostream>

using namespace std;

void find(int i, int n);

bool isprime(int x);

int main()
{
    int a,b;

    cin>>a>>b;

    find(a,b);

    return 0;
}

void find(int i, int n)
{
    int N=0;

    for(;N<n;i++)

    {

        if(isprime(i))
```

```

        {

            N+=1;

            cout<<i<<' ';

        }

    }

}

bool isprime(int x)
{

    if(x==1) return false;

    for(int i=2;i<x;i++)

    {

        if(x%i==0)

            return false;

    }

    return true;

}

```

### 2.3 运行截图

问题 输出 调试控制台 终端

> Executing task: D:\Codefield\CODE\_Cpp\Cpp\_Single\No.3\bin\2.exe <

13 4  
13 17 19 23  
按任意键关闭终端。

### 2.4 调试情况

Accepted, 见上图。

### 3、第三题

#### 3.1 题目描述

输入正整数 start 和 end，找出从 start 到 end 这一区间段内哪一个数 n 的因子和最大，并将该 n 及其因子和 maxSum 作为结果输出。例如，当 start=10, end=15 时，所求的 n 应该为 12，而 maxSum 应该为 16。

要求：设计函数计算 n 的因子和，其中因子和为包括 1 但不包括 n 本身的所有因子之和

#### 3.2 源代码

```
#include <iostream>

using namespace std;

int main()
{
    int start,end;

    int n,maxSum=0;

    cin>>start>>end;

    for(int i=start;i<=end;i++)
    {
        int tempSum=0;

        for(int j=1;j<i;j++)
        {
            if(i%j==0) tempSum+=j;

            if(tempSum>maxSum)
            {
                n=i;
            }
        }
    }
}
```

```

        maxSum=tempSum;

    }

}

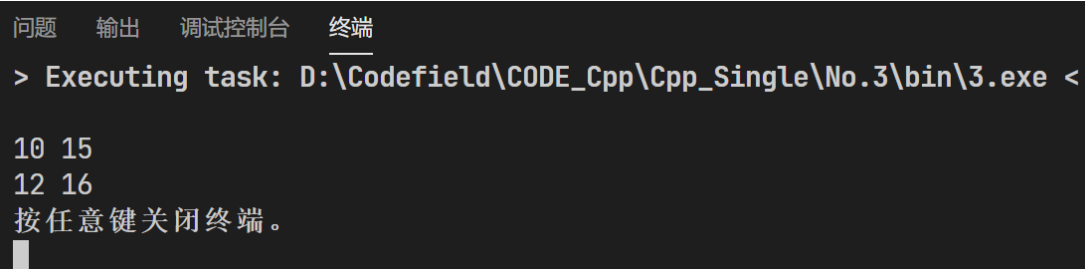
}

cout<<n<<' '<<maxSum;

return 0;
}

```

### 3.3 运行截图



```

问题  输出  调试控制台  终端
> Executing task: D:\Codefield\CODE_Cpp\Cpp_Single\No.3\bin\3.exe <

10 15
12 16
按任意键关闭终端。

```

### 3.4 调试情况

Accepted, 见上图。

## 4、第四题

### 4.1 题目描述

编写具有如下原型的函数 `symm`: `bool symm(long n)`; 用来判断正整数 `n` 是否为“回文数”（正读与反读为大小相同的数），若是返回 `true`，否则返 `false`。并编写主函数，通过调用 `symm`，求出 `n` 以内所有满足下述特征的 `m`: `m` 和 `7*m` 以及 `3*m*m` 都是“回文数”，如 1, 11, 88, ...（因为 `m=1`, `7*m=7`, `3*m*m=3`; `m=11`, `7*m=77`, `3*m*m=363`; `m=88`, `7*m=616`, `3*m*m=23232`; ...）

### 4.2 源代码

```

#include <iostream>

using namespace std;

bool symm(long n);

```

```
int main()
{
    int n;

    cin>>n;

    for(int m=1;m<=n;m++)

        if(symm(m)&&symm(7*m)&&symm(3*m*m)) cout<<m<<' ';

    return 0;
}

bool symm(long n)
{
    int p=0,q=n;

    while(q!=0)
    {
        p=p*10+q%10;

        q/=10;
    }

    if(p==n) return true;

    else return false;
}
```

#### 4.3 运行截图

问题 输出 调试控制台 终端

> Executing task: D:\Codefield\CODE\_Cpp\Cpp\_Single\No.3\bin\4.exe <

1000

1 11 88 101 111

按任意键关闭终端。

#### 4.4 调试情况

Accepted, 见上图。

### 5、第五题

#### 5.1 题目描述

键盘输入正整数  $n$ ，求出  $n$  与其反序数  $x$  之和并输出。例如，输入 2038， $n+x = 2038 + 8302 = 10340$ ，输出应为 10340。要求：编写函数实现数据转换成反序数值

#### 5.2 源代码

```
#include <iostream>

using namespace std;

long symm(long n);

int main()
{
    long n;

    cin>>n;

    cout<<n+symm(n);

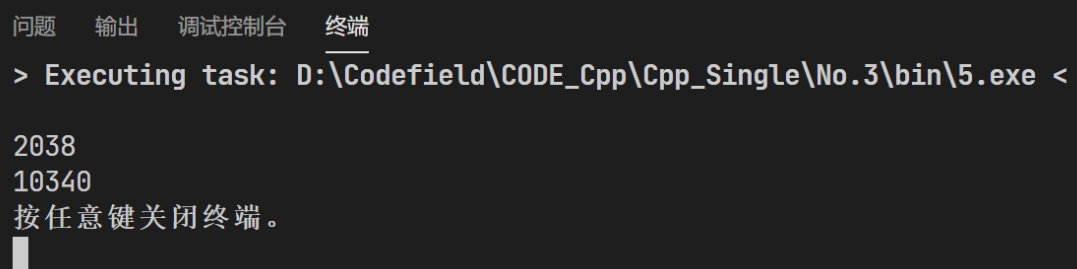
    return 0;
}

long symm(long n)
```



```
{  
  
    int p=0,q=n;  
  
    while(q!=0)  
    {  
  
        p=p*10+q%10;  
  
        q/=10;  
  
    }  
  
    return p;  
}
```

### 5.3 运行截图



```
问题  输出  调试控制台  终端  
> Executing task: D:\Codefield\CODE_Cpp\Cpp_Single\No.3\bin\5.exe <  
  
2038  
10340  
按任意键关闭终端。  
■
```

### 5.4 调试情况

Accepted, 见上图。

## 6、第六题

### 6.1 题目描述

给定一个十进制正整数 N，请将其转换为十六进制并输出。

### 6.2 源代码

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

```
void toHex(int n);

int main()
{
    int N;

    cin>>N;

    toHex(N);

    return 0;
}

void toHex(int n)
{
    if(n)
    {
        toHex(n/16);

        switch(n%16)
        {
            case 10:cout<<'A';break;

            case 11:cout<<'B';break;

            case 12:cout<<'C';break;

            case 13:cout<<'D';break;

            case 14:cout<<'E';break;

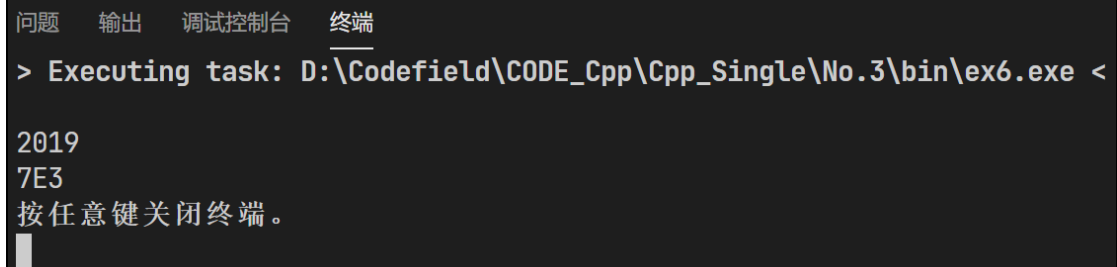
            case 15:cout<<'F';break;

            default:cout<<n%16;

        }
    }
}
```

```
}  
  
}
```

### 6.3 运行截图



```
问题  输出  调试控制台  终端  
> Executing task: D:\Codefield\CODE_Cpp\Cpp_Single\No.3\bin\ex6.exe <  
  
2019  
7E3  
按任意键关闭终端。  
█
```

### 6.4 调试情况

Accepted, 见上图。

### 三、实验体会

通过本次实验，巩固了函数的相关知识，并通过 CUMTOJ 这个平台的在线排行功能体会到了来自同学的速度压力和对程序设计的热情。