

## 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五(11月26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26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) 掌握指针变量的定义和初始化、指针的间接访问、指针的加减运算和指针表达式；			
2) 掌握用指针处理数组和字符串的方法；			
3) 掌握用指针作为函数参数的方法；			
4) 理解引用的概念，掌握引用的应用方法。			
二、实验内容			
1、第一题			
1.1 题目描述			
【题目描述】			
在进行文章重复度检查时，经常需要统计一段英文中的单词数量，并找出长度最长的单词。			
设有如下定义：char str[500];			
编写程序，通过利用 cin.getline(str, 500); 实现从键盘输入一小段英文（其中可以包含空格，但在同一行），利用函数统计该段英文中包含几个单词，输出统计出的单词数量、最长单词的长度以及长度最长的单词，空格隔开。			
注意：函数声明使用 void split(char *str); 如果有最长的单词不只一个，输出最先找到的那个。			
【输入】welcome to china university of mining and technology			

**【输出】** 8 10 university

## 1.2 源代码

```
#include <iostream>

using namespace std;

void split(char *str);

int main()
{
    char str[500];

    cin.getline(str, 500);

    split(str);

    return 0;
}

void split(char *str)
{
    int i, j, t=0, l[20], max=0, m=0;

    for(i=0;*(str+i);i++)
    {
        if(*(str+i)==' ')
        {
            l[t]=i;

            t++;
        }
    }
}
```

```

    l[t]=i;

    max=l[0];
    for(j=0;j<t;j++)
    {
        if(max<(l[j+1]-l[j]))
        {
            max=l[j+1]-l[j];
            m=j;
        }
    }

    cout<<t+1<<' ' <<max-1<<' ';

    for(i=l[m]+1;i<l[m+1];i++)

        cout<<*(str+i);
}

```

### 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.5\bin\1.exe <

welcome to china university of mining and technology
8 10 university
按任意键关闭终端。

```

### 1.4 调试情况

Accepted, 如上图。

## 2、第二题

### 2.1 题目描述

#### 【题目描述】

编程序，按如下方法求 A 矩阵的转置矩阵 B：输入两个正整数 m 和 n，而后通过使用指针配合 new 运算符生成一个 m 行 n 列的二维动态数组 A 以及另一个 n 行 m 列的二维动态数组 B，之后为 A 输入数据（A 矩阵数据），逐行逐列输入，进而求出其转置矩阵 B（即将 A 中的行存放成 B 中的列）并输出结果，逐行逐列输出，每一行数字占一行，同一行的数空格隔开。

注意：输入保证全是整数。

#### 【输入】

3 2

119 65

629 679

721 564

#### 【输出】

119 629 721

65 679 564

### 2.2 源代码

```
#include <iostream>

using namespace std;

int main()
{
    int m,n,i,j;

    cin>>m>>n;

    int (*p)[100]=new int[100][100];
```

```

    for(i=0;i<m;i++)

        for(j=0;j<n;j++)

            cin>>*(*(p+i)+j);

    for(j=0;j<n;j++)

    {

        for(i=0;i<m;i++)

            cout<<*(*(p+i)+j)<<' ' ;

        cout<<endl;

    }

    return 0;
}

```

### 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.5\bin\2.exe <

3 2
119 65
629 679
721 564
119 629 721
65 679 564

```

### 2.4 调试情况

Accepted, 如上图。

## 3、第三题

### 3.1 题目描述

编写如下原型的函数：

**【题目描述】**

```
void split(double x,int*iPart,double*fPart);
```

提取出数据 x 的整数部分与小数部分，分别放于 iPart 与 fPart 处，由于形参 iPart 与 fPart 都是指针，从而可实现将这两个结果“带回”到主函数中。

在主函数中输入一个数输出它的整数部分和小数部分，用空格隔开。

提示：一个 double 类型数，强制类型转换后就是 int，也就是整数部分，差为小数部分。这两个值用指针 iPart 和指针 fPart 带回（通过修改指针的目标变量值。）

**【输入】** 12.3

**【输出】** 12 0.3

### 3.2 源代码

```
#include <iostream>

using namespace std;

void split(double x,int*iPart,double*fPart);

int main()
{
    double x,b,*f;

    int a,*i;

    cin>>x;

    f=&b;

    i=&a;

    split(x,i,f);

    cout<<a<<' '<<b;

    return 0;
}
```

```
void split(double x,int*iPart,double*fPart)
{
    int a;

    double b;

    a=x;

    b=x-a;

    *iPart=a;

    *fPart=b;
}
```

### 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.5\bin\3.exe <
12.3
12 0.3
按任意键关闭终端。
```

### 3.4 调试情况

Accepted, 如上图。

## 4、第四题

### 4.1 题目描述

#### 【题目描述】

编制具有如下原型的函数 findLast:

```
char *findLast(char*sourceStr,char*subStr);
```

findLast 函数则要返回源串 sourceStr 中最后一次出现 subStr 子字符串的头字符位置。而后编制主函数, 输入两个字符串, 将它们用作实参来调用这两个

函数，如果返回 NULL 输出-1，否则输出子字符串出现时头字符在原字符串的下标，每个结果占一行。

要求实现程序中不可使用“string.h”头文件内有关寻找子串的标准库函数。

**【输入】**

welcometochinauniversityofminingandtechnology

in

**【输出】** 29

#### 4.2 源代码

```
#include <iostream>
```

```
using namespace std;
```

```
char *findLast(char*sourceStr, char*subStr);
```

```
int main()
```

```
{
```

```
    char a[500], b[50], *p;
```

```
    cin>>a;
```

```
    cin>>b;
```

```
    p=findLast(a, b);
```

```
    if(p==NULL)
```

```
        cout<<-1;
```

```
    else cout<<p-a;
```

```
    return 0;
```

```
}
```

```
char*findLast(char*sourceStr, char*subStr)
```

```
{
```



```
int x=0, y=0, i, j, t=0;

while(*(sourceStr+x))

    x++;

x--;

while(*(subStr+y))

    y++;

y--;

for(i=x; i>=0; i--)

{

    if(*(sourceStr+i)==*(subStr+y))

    {

        t=0;

        for(j=0; j<=y; j++)

            if(*(sourceStr+i-j)!=*(subStr+y-j))

                t++;

        if(t==0)

        {

            return sourceStr+i-j+1;

            break;

        }

    }

}

return NULL;
```

```
}
```

#### 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.5\bin\4.exe <

welcometochinauniversityofminingandtechnology

in

29

按任意键关闭终端。

□

#### 4.4 调试情况

Accepted, 如上图。

### 5、第五题

#### 5.1 题目描述

##### 【题目描述】

编写具有如下原型的函数：int f(unsigned long x, int n, int& Lxn);

它负责将整数 x 的第 n 位（从左边数第 n 位，n>0）的数值放到引用 Lxn 之中（将作为结果返回到主调函数的对应实参变量中），并将倒数第 n 位（从右边数第 n 位，n>0）的数值作为函数结果返回去。并编制主函数对它进行调用以验证其正确性。例如，当 x=123456789，n=7 时，执行语句“Rxn=f(x, n, Lxn);”将使返回的 Lxn 为 7，并使 Rxn 变为 3；而执行语句“Rxn=f(12345, 6, Lxn);”将使 Lxn 与 Rxn 都变为 0（超出数的“长度”即总位数时返回 0）

##### 【输入】

输入 x, 输入 n

##### 【输出】

输出 Rxn, 输出 Lxn

#### 5.2 源代码

无

### 5.3 运行截图

无

### 5.4 调试情况

Wrong Answer

### 三、实验体会

通过本次实验，巩固了指针与引用的相关知识，并通过新的 CUMTOJ 平台的在线排行功能体会到了自身实力的不足以及来自同学的速度压力和对程序设计的热情。