



**程序设计实验报告2**

学 院

专 业

年级班别

学 号

学生姓名

指导教师

2018年 10 月

# 实验2

## 实验目的

1. 理解基本的语句，如控制语句、赋值语句等；
2. 能够结合输入输出语句设计编写简单的程序；
3. 在了解关系运算符和关系表达式，逻辑运算符和逻辑表达式的基础上，能够编写具有选择结构的程序；
4. 能够区别多种循环，具备实现循环的能力，编写具有循环嵌套的程序。

## 实验工具

Tubro C 2.0/3.0

## 编写要求

对每个实验题目，需要提供如下内容：

1. 程序流程图（应采用NS图或传统流程图）；
2. 具体程序代码（具有良好的程序设计风格，如程序模块化、加注释、缩进的书写格式）；
3. 实验运行结果（对比测试用例）；
4. 对运行情况所做的分析以及调试程序所取得的经验收获。

## 排版要求

1. 标题四号宋体，正文小四宋体，段间距1.5倍；
2. 除封面以外，其他内容双面打印。

## 实验题目

1． 接收一个六位正整数。编写一个程序，将该数的每一位数字相加并显示结果，最后将这六位数从大到小排序输出，比如314500，输出和13，排序后成为一个新的数543100。若输入的不是六位数，则要提示出错。用循环实现。

测试用例：(1)123456 (2)890100 (3)12

**NS图：**

**fun函数**

long int fun(int \*a, int n, int x)

|  |  |
| --- | --- |
| Y  n <= 1  N | |
| return a[0] + fun(a + 1 , n - 1, x) \* x; | return a[0]; |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| long int num; | | | | |
| Int a[6],sum=0,i=0,j,k,temp; | | | | |
| 输入num | | | | |
| num >= 100000 && num < 1000000  N  Y | | | | |
| while(num) | | | | 输入错误 |
|  | a[i] = num % 10; | | |
| num /= 10; | | |
| sum += a[i++]; | | |
|  | for (i = 0; i < 5; i++) | | |
|  | k=i | | |
| for (j = i + 1; j < 6; j++) | | |
|  | Y  a[k]<a[j] | N |
| k=j |  |
|  | N  Y  k!=i | |
| temp=a[i]  a[i]=a[k]  a[k]=temp | |  |
| num=fun(a,6,10); | | | |
| 输出sum，num。 | | | | |

**main函数**

**代码：**

#include <stdio.h>

long int fun(int \*a, int n, int x)//用递归使num中的数从大到小排序

{

if (n <= 1)

return a[0];

else

return a[0] + ( fun (a + 1 , n - 1, x)) \* x;

}

int main()

{

long int num;

int a[6], sum = 0, temp, i = 0, j, k;

printf("Please enter an integer number: ");

scanf("%ld", &num);

if (num >= 100000 && num < 1000000)

{

while (num)//将位数分离

{

a[i] = num % 10;

num /= 10;

sum += a[i++];

}

for (i = 0; i < 5; i++)

{//将位数排序

k = i;

for (j = i+1; j < 6; j++)

if (a[k] > a[j])

k = j;

if (k != i)

{

temp = a[i];

a[i] = a[k];

a[k] = temp;

}

}

num=fun(a, 6, 10);

printf("%d\n", sum);

printf("%ld\n", num);

}

else

{

fprintf(stderr, "Improper value of num\n");

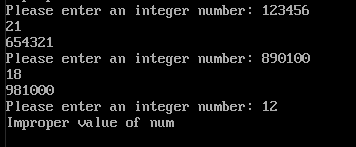
exit(1);

}

return 0;

}

**运行结果：**

**总结：**

这道题如果要从屏幕直接输出一道位数从大到小的数是可以直接输出一个数组的，但是那样只是在屏幕看上去是一个数，其实是多个元素拼接起来的。所以为了让num按题目那样输出，我写了一个递归函数，通过递归实现了num的转变。