天津大学

数据结构上机实验报告

题目:根据关键字分解字符串

学生姓名 王雨朦

学生学号 2016229082

学院名称 国际工程师学院

专业 计算机

时间 2016/10/23

目　　录

第一章　需求分析

1.1 原题表述

输入一行数字，如果我们把这行数字中的‘5’都看成空格，那么就得到一行用空格分割的若干非负整数（可能有些整数以‘0’开头，这些头部的‘0’应该被忽略掉，除非这个整数就是由若干个‘0’组成的，这时这个整数就是 0）。

具体要求是：对这些分割得到的整数，依从小到大的顺序排序输出。

1.2 解决方案

1) 可以循环输入字符串；
2) 对每一个输入的字符串，对字符串进行分割；
3) 先判断是否遇到 5。如果是 5 则递归往下继续搜索；如果不是则将字符串转化为数字存入数字字符串中，以便后面比较大小；
4) 将存入数字数组中的数字进行升序排序；
5) 输出数字数组。

第二章　概要设计

2.1 抽象数据类型

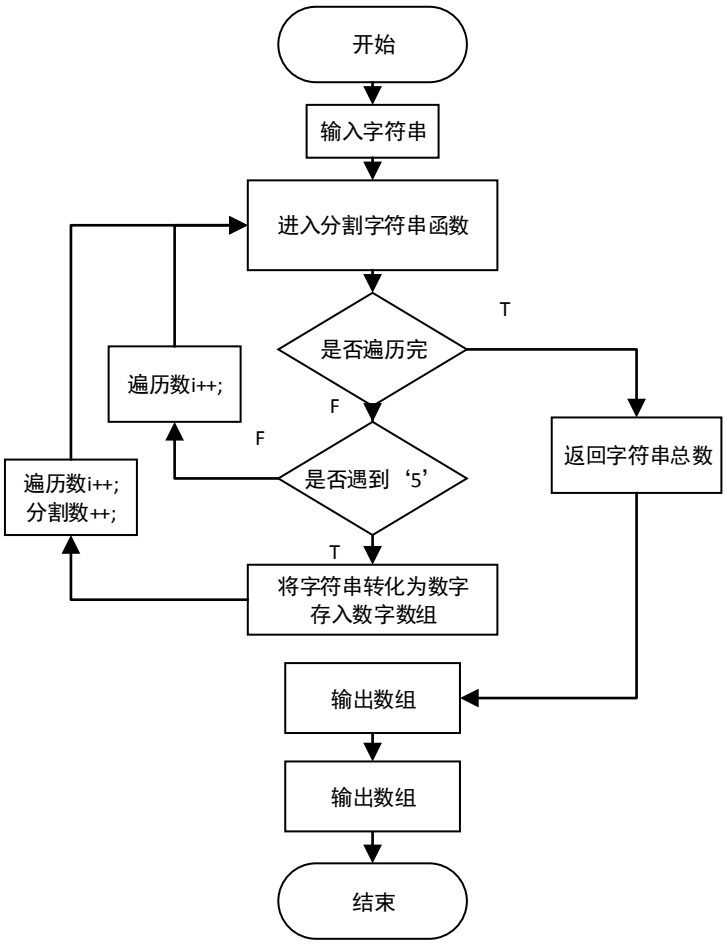
整形数组和字符串:

```
#define MAXSTRLEN 1001
int ans[MAXSTRLEN];
char str[MAXSTRLEN];
```

分解字符串函数（递归）:

```
int destr(int i,int num,int len,char str[])
void display(int ans[],int len)
```

2.2 算法



第三章　详细设计

3.1 程序代码

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <string.h> //strlen()
#include<algorithm> //sort
#define MAXSTRLEN 1001
using namespace std;
int ans[MAXSTRLEN];
int destr(int i,int num,int len,char str[]){
	//printf("i:%d num:%d len:%d \n",i,num,len);
	if(i >= len) return num;//终止条件
	if(str[i]=='5') return destr(i+1,num,len,str);//以‘5’开头或者连续两个 5
	int sum=0;
	while(str[i]!='5' && i<len){
		sum*=10;
		sum+=str[i]-'0';
		i++;
	}
	ans[num]=sum;
	return destr(i+1,num+1,len,str); //递归下一个分割串
}
void display(int ans[],int len){
	for(int i=0;i <len;i++){
		cout<<ans[i]<<" ";
	}
	cout<<endl;
}
int main(){
	char str[MAXSTRLEN];
	int len,numstr;
	printf("Please input a string('5' is the segmentation):\n");
	while(cin>>str){
		len=strlen(str);
		numstr=destr(0,0,len,str);//i 是计数器计算遍历到第 i 个字符；num 是分解到的字符串数目；
		sort(ans,ans+numstr);
		display(ans,numstr);}
}
```

第四章　调试分析

4.1 调试过程

Bug 名称	逻辑错误
Bug 描述	最后一个字符如果不是 5，就无法输出
Bug 原因	递归函数中终结条件没有放在最开始
Bug 解决方案	递归条件放在了最开始
Bug 总结	递归函数用的太少

Bug 名称	函数错误
Bug 描述	进入分割函数后程序无法结束
Bug 原因	递归函数最后一行没有递归条件
Bug 解决方案	在最后一行加上递归条件
Bug 总结	递归函数用的太少

Bug 名称	边界错误
Bug 描述	输出的分割字符串总是少一个
Bug 原因	输出函数的边界条件错从 1 开始了
Bug 解决方案	更正边界条件
Bug 总结	进行代码编写时，思路不够清晰

Bug 名称	逻辑错误
Bug 描述	当输入第二个字符串时，输出值总不对
Bug 原因	初始化的值错放在了最开始
Bug 解决方案	将初始化语句放在循环开始
Bug 总结	思路不清晰，写循环之前没有认真思考

第五章　测试结果

5.1 测试过程

测试编号	1
测试对象	5 在开头时
测试输入参数	589757857
测试步骤	输入以 5 开头的字符串，观察输出
测试预期结果	7 78 897
测试输出结果	7 78 897
测试分析	程序正确

```
Please input a string('5' is the segmentation):
589757857
7 78 897
```

测试编号	2
测试对象	5 在结尾且全 0 时
测试输入参数	5789598750005
测试步骤	输入全 0 且以 5 结尾的字符串，观察输出。
测试预期结果	0 789 987
测试输出结果	0 789 987
测试分析	程序正确

```
5789598750005
0 789 987
```

测试编号	3
测试对象	带 0 的数字
测试输入参数	09875908759870
测试步骤	输入开头中间结尾分别有 0 的字符串，观察输出。
测试预期结果	987 9087 9870
测试输出结果	987 9087 9870
测试分析	程序正确

```
09875908759870
987 9087 9870
```