

计算机导论与程序设计 [CS006001-60]

段江涛

机电工程学院



2019 年 11 月

lecture-16 主要内容

指针综合应用举例, 期中考试总结

- 1 指针综合应用举例
- 2 期中考试总结
- 3 整数 pow 函数问题

例: 表达式求值

表达式由两个非负整数 x, y 和一个运算符 op 构成, 求表达式的值。

这两个整数和运算符的顺序是随机的, 可能是 “ $x \text{ op } y$ ”, “ $op \ x \ y$ ” 或者 “ $x \ y \ op$ ”, 例如, “ $25 + 3$ ” 表示 25 加 3, “ $5 \ 30 \ *$ ” 表示 5 乘以 30, “ $/ \ 600 \ 15$ ” 表示 600 除以 15。

输入说明

输入为一个表达式, 表达式由两个非负整数 x, y 和一个运算符 op 构成, x, y 和 op 之间以空格分隔, 但顺序不确定。 x 和 y 均不大于 10000000, op 可以是 $+, -, *, /$, $\%$ 中的任意一种, 分表表示加法, 减法, 乘法, 除法和求余。除法按整数除法求值, 输入数据保证除法和求余运算的 y 值不为 0。

输出说明

输出表达式的值。

输入样例

样例 1 输入

5 20 *

样例 2 输入

4 + 8

样例 3 输入

/ 8 4

输出样例

样例 1 输出

100

样例 2 输出

12

样例 3 输出

2

例：表达式求值—解题思路

- gets函数读取字符串,遍历字符串,根据 op 字符是”非数字字符”的特点,判断表达式的三种形式。
- 设计各个子函数,进行模块化程序设计。
- 编写函数 parse,解析输入字符串,生成3个子串,分别代表 op,x,y
- 编写函数 strToInt,将数字字符串转为整数。
- 编写函数 compute,根据 op,x,y 计算表达式的值。
- 编写其它辅助函数。
- 主函数,调用上述函数,完成程序功能。

注:

`int s1[81],s2[81],s3[81]; scanf("%s%s%s",s1,s2,s3);` // 遇空格结束特点(忽略前缀后缀空格),自动获取3个子串,无需分解。

例：表达式求值—计算函数

// 根据参数，计算表达式的值

```
int compute(char op,int x,int y)
{
    int result = -1;
    switch(op)
    {
        case '+': result = x+y; break;
        case '-': result = x-y; break;
        case '*': result = x*y; break;
        case '/': if(y != 0) result = x/y; break;
        case '%': if(y != 0) result = x%y; break;
    }
    return result;
}
```

例：表达式求值——将数字字符串转为整数函数

```
// 数字字符串s转为int, 要求s以'\0'结尾
int strToInt(char *s) //等效: int toInt(char s[])
{
    int result=0;
    while(*s) //等效 *s != '\0'
    {
        result = result*10 + (*s-'0');
        s++; //移至下一字符
    }
    return result;
}
```

注：简单修改此函数, 可将由 01 组成的二进制字符串转为整数。

例：表达式求值—提取子串函数

```

/*****
提取子串函数，忽略s中空格外缀，复制s中的字符串到subs中，遇空格或'\0'结束
返回subs不含空格。返回复制后s指针当前指向(地址)，要求s和subs以'\0'结尾。
*****/

char* getSubs(char *s, char *subs)
{
    int start=0; // 是否开始提取标志
    while(*s)
    {
        if(*s=='\u')
        {
            if(start==0) s++; // 未开始复制，忽略s的前缀空格
            else break; // 是有效字符串后的一个空格
        }
        else
        {
            start=1; // 标记开始复制
            *subs=*s; s++; subs++;
        }
    }
    *subs='\0'; // 不要忘记结尾符
    return s; // 返回复制后s指针当前指向(地址)
}

```

注: 简单修改此函数, 可以任何分隔符提取子串函数

例：表达式求值—解析字符串函数

```
// 解析s，以空格为分隔符，分解s为3个字符串，通过参数result返回  
void parse(char *s, char result[][N])  
{  
    char *p;  
    p=getSubs(s, result[0]);  
    p=getSubs(p, result[1]);  
    getSubs(p, result[2]);  
}
```

注：通过此函数，学习字符串数组的使用技巧。

例: 表达式求值—是否运算符函数

// 如果s是操作符, 返回1, 参数op返回该操作符

// 否则, 返回0

```
int isOp(char *s, char *op)
{
    if(*s >= '0' && *s <= '9')
        return 0;
    else
    {
        *op=*s;
        return 1;
    }
}
```

注: 通过此函数, 学习地址传递技巧。

例：表达式求值—测试主函数

```
int main1() // 测试主函数，使能改为main()，原main()改名
{
    char *s="123";
    printf("%d\n",strToInt(s)); // 123
    char s1[N]="uuuuabcduu456++", s2[50], *s3;
    s3= getSubs(s1,s2);
    printf("%s,%d\n",s2,*s3); // abcd,32 (空格的ASCII码)
    s3= getSubs(s3,s2);
    printf("%s,%d\n",s2,*s3); // 456,32 (空格的ASCII码)
    s3= getSubs(s3,s2);
    printf("%s,%d\n",s2,*s3); // ++,0 ('\0'的ASCII码)
    char result[3][N];
    parse(s1,result);
    puts(result[0]); // abcd
    puts(result[1]); // 456
    puts(result[2]); // +
    return 0;
}
```

注: main() 和 main1() 相互换名, 调试程序。是实用调试程序技巧。

例：表达式求值—主程序

```
#include <stdio.h>

// 自定义函数在调用之前定义，略去函数声明。

#define N 81 // 估计字符串最大长度，存储有效字符 (N-1) 个，预留最后一个字符'\0'

int main()
{
    char s[N], op, s3[3][N];
    int x, y;

    gets(s); // 不能使用scanf("%s", s); 空格将会终止
    parse(s, s3); // s被分解为3个字符串
    if(isOp(s3[0], &op)) // op x y
    {
        x=strToInt(s3[1]); y=strToInt(s3[2]);
    }
    else if(isOp(s3[1], &op)) // x op y
    {
        x=strToInt(s3[0]); y=strToInt(s3[2]);
    }
    else if(isOp(s3[2], &op)) // x y op
    {
        x=strToInt(s3[0]); y=strToInt(s3[1]);
    }
    printf("%d\n", compute(op, x, y));
    return 0;
}
```

例：表达式求值—主程序 (另解)

```
#include <stdio.h>

// 自定义函数在调用之前定义，略去函数声明。

#define N 81 // 估计字符串最大长度，存储有效字符 (N-1) 个，预留最后一个字符'\0'

int main()
{
    char s[N], op, s3[3][N];
    int x, y;
    scanf("%s%s%s", s3[0], s3[1], s3[2]); // 利用"%s"读字符串遇空格结束特点，直接
        读取3个字符串。
    if(isOp(s3[0], &op)) // op x y
    {
        x=strToInt(s3[1]); y=strToInt(s3[2]);
    }
    else if(isOp(s3[1], &op)) // x op y
    {
        x=strToInt(s3[0]); y=strToInt(s3[2]);
    }
    else if(isOp(s3[2], &op)) // x y op
    {
        x=strToInt(s3[0]); y=strToInt(s3[1]);
    }
    printf("%d\n", compute(op, x, y));
    return 0;
}
```

期中考试总结

1 评分规则

- 将 5 个题的得分按从高到低排序, 记为 S1, S2, S3, S4, S5
- 总分 = $S1*0.30+S2*0.25+S3*0.20+S4*0.15+S5*0.1$

2 第 1 题: 自由落体, 基本计算, 仅有 1 人没得满分。

3 第 2 题: 运费: 5 人未得满分, 3 人 0 分

与 lecture-6.pdf 中的例题基本一致。考查 switch 语句或 if else 选择语句。

4 第 3 题: 二进制字符转换, 考查一重循环, 本题平均成绩最低 44。

lecture-8.pdf p15-16, 循环语句中接收字符的常见技巧。

lecture-5.pdf p8, 数字 = ASCII 编码值 - '0'

lecture-14.pdf p20, 以递归形式输出一个整数的二进制位。

课件及上机练习中 (如 ISBN) 涉及各种数字分解程序设计技巧。

5 第 4 题, 考查一重循环, 平均成绩 86。

lecture-9.pdf p3 原题, 迭代计算是基本编程要领, 必须掌握。

6 质数求和, 考查二重循环, 平均成绩 51。

lecture-8.pdf p8—p11, 100~200 间的素数。

7 本班题目简单, 其他班有类似机试练习中选 QQ 号题。期末考试会复杂些。

第 3 题—二进制字符转换, 简单实现

10101d \Rightarrow 21 101,1 \Rightarrow 5

```
#include<stdio.h>

int main()
{
    char ch;
    int sum=0;
    while(1)
    {
        ch=getchar(); //或 scanf("%c",&ch);
        if(ch!='0' && ch !='1') break;
        sum=sum*2+ch-'0';
    }
    printf("%d\n",sum);
    return 0;
}
```

第 3 题—二进制字符转换, 字符串数组实现

```
#define N 31 // 估计最大数组大小, 预留'\0'
char ch[N];
int sum=0,i;
gets(ch);
for(i=0;;i++)
{
    if(ch[i]!='0' && ch[i] !='1') break;
    sum=sum*2+ch[i]-'0';
}
printf("%d\n",sum);
```

第 3 题—二进制字符转换, 指针实现

```
#define N 31 // 估计最大数组大小, 预留'\0'
char ch[N], *p=ch;
int sum=0;
gets(ch);
for(;;)
{
    if(*p != '0' && *p != '1') break;
    sum=sum*2+ (*p - '0');
    p++; // 指向下一个字符
}
printf("%d\n", sum);
```


整数 pow 函数问题

`double pow(double x, double y)`; 数学库函数真对双精度浮点数设计。用它计算整数 x^y 会有复杂的精度问题, 因此整数运算尽量不要用此函数。

```
#include<stdio.h>

#include<math.h>    // 数学库函数

int main()
{
    int x=2,y=3;

    // 有些编译器会输出99,124, 因为转换前是99.999, 124.999
    printf("%d,%d\n", (int)pow(10,x), (int)pow(5,y));

    // 奇怪的是下列调用输出结果与上面有可能会输出不一致的结果
    printf("%d,%d\n", (int)pow(10,2), (int)pow(5,3));

    return 0;
}
```

自定义整数 pow 函数

```
#include<stdio.h>
// 自定义整数pow函数
int intpow(int x, int y)
{
    int i,p=1;
    if(x==0) return 0;
    for(i=0;i<y;i++) p=p*x;
    return p;
}

int main()
{
    int x=2,y=3;

    printf("%d,%d\n",intpow(10,x),intpow(5,y)); // 100,125
    printf("%d,%d\n",intpow(10,2),intpow(5,3)); // 100,125
    printf("%d,%d\n",intpow(0,2),intpow(0,0)); // 0,1

    return 0;
}
```

欢迎批评指正！