

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-15 主要内容

指针应用题, 期中考试总结

- 1 指针应用题
- 2 期中考试总结

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。

注意: 因为字符串中含有空格,不能使用 `scanf("%s", -)`; 读取字符串。

- 编写计算函数, 根据 `op,x,y` 计算表达式的值。
- 编写独立函数, 提取 `[*p1,*p2]` 之间的数字字符串, 并返回对应的整数值 (`x` 或 `y`)。
- 主程序, 解析字符串, 调用上述函数。

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—提取字符串中的数字

// 提取[*p1,*p2]之间的数字字符串，并返回对应的整数值(xy)。

```
int xy(char *p1,char *p2)
{
    int result = 0;
    // 删除前后缀空格
    while(*p1 == ' ') p1++; // 使p1,p2指向非空格
    while(*p2 == ' ') p2--;
    for(; p1 <= p2; p1++)
    {
        result += *p1 - '0'; // 1 = '1' - '0'
        if(p1 != p2) result *= 10; // 非最后一个字符，左移一位十进制数
    }
    return result;
}
```

例：表达式求值—主程序

```
#include <stdio.h>

// 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'
#define N 20

// 根据参数，计算表达式的值
int compute(char op,int x,int y); // 函数声明

// 提取[*p1,*p2]之间的数字字符串，并返回对应的整数值(xy)。
int xy(char *p1,char *p2);

int main()
{
    char s[N],*p1,*p2,op; // 借助p1,p2指针变量，扫描字符串
    int x,y;
    gets(s); // 不能使用scanf("%s",s); 空格将会终止
    p1 = s; // p1指向字符串首地址
    while(*p1 == ' ') p1++; // 使p1指向非空格
    if (*p1 < '0' || *p1 > '9') // 首先判断第一个字符是运算符, op x y
    { 续1 }
    else // x y op 或 x op y
    { 续2,续3}
    printf("%d\n",compute(op,x,y));
    return 0;
}
```

例: 表达式求值—主程序 (续 1)

```
// 首先判断第一个字符是运算符, op x y
if (*p1 < '0' || *p1 > '9')
{
    op = *p1;
    p1++; // 掠过op
    while(*p1 == ' ') p1++; // 使p1指向非空格
    p2 = p1;
    while(*p2 != ' ') p2++; // x空格
    x = xy(p1, p2);
    p1 = p2;
    while(*p2 != '\0') p2++; // p2指向'\0'
    y = xy(p1, p2 - 1); // p2 - 1是最后一个有效字符
}
```


例：表达式求值—主程序 (续 2)

```
else // x y op 或 x op y
{
    p2 = s; // p2指向字符串首地址
    while(*p2 != '\0') p2++; // p2指向'\0'
    p2--;
    while(*p2 == ' ') p2--; // p2指向非空格
    if (*p2 < '0' || *p2 > '9') // x y op
    {
        op = *p2;
        p2--; // 掠过op
        while(*p2 == ' ') p2--; // p2指向非空格
        p1 = p2;
        while(*p1 != ' ') p1--; // 空格y
        y = xy(p1, p2);
        x = xy(s, p1);
    }
    else // x op y
    { 续3 }
}
```

例：表达式求值—主程序 (续 3)

```
else // x op y
{
    p1 = s;
    while(*p1 == ' ') p1++; // 使p1指向非空格
    p2 = p1;
    while(*p2 != ' ') p2++; // x空格
    x = xy(p1,p2);
    while(*p2 == ' ') p2++; // op
    op = *p2;
    p1 = p2 + 1; // 掠过op
    p2 = p1;
    while(*p2 != '\0') p2++; // p2指向'\0'
    y = xy(p1,p2 - 1); // p2 - 1是最后一个有效字符 ^^I
}
```

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 数位 = ASCII 编码值 - '0'

lecture-14.pdf 以递归形式输出一个整数的二进制位。

课件及上机练习中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 p24 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, p17 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

```
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);
```

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);
```

欢迎批评指正！