

C 语言算法总结 (1-6)

一、表达式

1 “/” 和 “%” 的用法

/:用来降阶 %: 用来求余, 一般是求个位数。

例子: 已知一五位数 n , 求个位, 十位, 百位, 千位, 万位是多少?

个位: $n\%10$, 十位: $n/10\%10$, 百位: $n/100\%10$, 千位 $n/1000\%10$, 万位: $n/10000$

用途: 用在水仙花, 完数等题目。实验指导 p168 页 8 题。

2 整除的表达式

例子: i 能被 2 整除又能被 3 整除。

$i\%2==0 \ \&\& \ i\%3==0$

3 闰年的表达式。

例子: 判断 year 是否是闰年。

$year\%4==0 \ \&\& \ year\%100!=0 \ || \ year\%400=0$

4 ++ --

++p: 表示先让 $p+1$, 再把 $p+1$ 作为 p 的值进行使用, 即先加后用。

P++: 表示先使用 p 的值, 使用结束后, 再把 $p+1$ 作为 p 的值, 即先用后加。

5 if (exp) 的说明。

此处与 exp 等价的是 $exp!=0$ 。

If(4), if(-2) 等都表示条件满足。

6 关于 “double” 数据

已有定义: `double a,b;`

输入: `scanf("%lf",&a);`

输出: `printf("%f",a);`

7 关于相等:

整型相等: a, b 为整型 $a==b$

double 类型相等: $\text{fabs}(a-b)\leq 1e-6$

字符串相等: `strcmp(a,b)==0`

8 关于赋值:

整型或 double 型变量之间赋值。 $a=b$;

字符串之间赋值 (字符串复制): `strcpy (a,b) ;`

二、程序片段

1 确保 n 为正数:

```
do
{
    printf("Enter n: ");
    scanf("%d", &n);
} while (n <= 0);          /* 确保 n 为正数 */
```

同类算法, 确保 n 在 1-10 之间。

```
do
{
    printf("Please input n(1<=n<=10):\n");
    scanf("%d",&n);
}while (n<1 || n>10);     /* 确保 n 在 1-10 之间 */
```

2 一维数组的元素输入、输出

```
int score[5],i;
for(i=0;i<5;i++)
scanf("%d",&score2[i]);    换成 “printf(“%d”,score2[i]);” 则为输出。
```

3 二维数组的元素输入、输出

```
int a[2][3];
int i,j;
for(i=0;i<2;i++)
for(j=0;j<3;j++)
scanf("%d",&score2[i][j]);    换成 “printf(“%d”,score2[i][j]);” 则为输出。
或采用宏定义的方式,好处是如果要改变行数或列数, 只需要改变一个地方。
```

```
#define ROW 2
#define COL 2
int a[ROW][COL];
int i,j;
for(i=0;i<ROW;i++)
for(j=0;j<COL;j++)
scanf("%d",&score2[i][j]);
```

控制每行打印个数, 例如每行输出 5 个。

```
int score[5],i,count=0;
for(i=0;i<5;i++)
{
printf("%d",score2[i]);
count++;
if(count%5==0)
printf(“\n”);
}
```

4 数组元素查找 p112

```
int find(int a[],int n,int x)
{
    int i=0;
    while(i<n)                /*循环条件为: 如果未找到且未搜索完元素*/
    {
        if (x==a[i])          /*如果查找成功, i 的值正好是元素下标*/
            break;
        i++;
    }
    return i;
}
```

5 数组元素插入 p115

```

void insert(int a[],int n,int x)
{
    int i,j;
    for (i=0;i<n&& a[i]<x;i++);          /*定位：查找待插入的位置 i，循环停止时的 i 就是
*/
    for (j=n-1;j>=i;j--)                /*移位：用递减循环移位，使 i 下标元素可被覆盖*/
        a[j+1]=a[j];
    a[i]=x;                             /*插入：数组的 i 下标元素值赋值为插入的 x*/
}

```

6 数组元素删除 p116

```

int delArray(int a[],int n,int x)
{
    int i,j;
    int flag=1;                          /*是否找到待删元素的标志位，1 找到，0
未找到*/
    for (i=0;i<n && a[i]!=x;i++);        /*查找 x 是否存在，此处循环体为空语句*/
    if (i==n)                             /*循环停止时如果 i==n，则说明元素不存在
*/
        flag=0;
    else
    {
        for (j=i;j<n-1;j++)
            a[j]=a[j+1];                 /*前移覆盖 i 下标的元素*/
    }
    return flag;
}

```

7 冒泡法排序 p118

```

void BubbleSort(int a[], int n)
{
    int i, j,temp;
    for (i = 0; i < n-1; i++)            /*共进行 n-1 趟排序*/
        for (j = n-1; j > i; j--)        /*递减循环，从后往前比较，趟号增加，相邻元素对
减少*/
            if (a[j] < a[j-1])           /*两两比较，若后一个元素小则交换该组相邻元素*/
            {
                temp=a[j-1];
                a[j-1]=a[j];
                a[j]=temp;
            }
}

```

三、完整程序

1 数列求和： p61 例 4.8。同类题目 P70 页 6。实验指导 p163 页 3、5、6

```
#include <stdio.h>

int main( )
{
    int i, sign;
    double item, sum;

    sum = 0; /* 初值置为 0 */
    sign = 1;

    for ( i = 1 ; i <= 1000 ; i++ )
    {
        item = sign / ( 2.0 * i - 1 ); /* 计算每一次的累加项 item */
        sum += item; /* 将累加项 item 加到总和 sum 上 */
        sign = -sign; /* 计算下一个累加项的符号 sign */
    }

    printf( "sum = %f\n", sum );
    return 0;
}
```

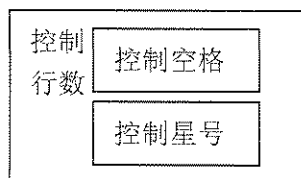
2 图形打印。p65 例 4.10。同类题目 P70 页 7。

图形打印一般都是利用 for 循环嵌套完成，外层循环控制行，内层循环控制列。

例如如下图形：

```
***
*****
*****
*****
```

(图 1)



(图 2)

分析：本图形共有四行，所以外层循环控制变量从 i 到 4，对每一行来说，他由空格、星号和回车构成，空格和星的数量不等，因此需要不同的循环，分别控制空格和星号。如图 2。

```
#include <stdio.h>
```

```
int main( )
{
    int i, j;
    for ( i = 1 ; i <= 4 ; i++ ) /* 外层循环：控制行数 */
    {
        for ( j = 1 ; j <= 4-i ; j++ ) /* 内层循环：控制空格数 */
        {
            printf( " " ); /* 输出空格 */
        }
        for ( j = 1 ; j <= 2*i+1 ; j++ ) /* 内层循环：控制星号数 */
        {
```

```

        printf( "*" );                /* 输出星号 */
    }
    printf( "\n" );                    /* 输出回车 */
}
return 0;
}

```

3 判断质数：p65 例 4.12 同类题目 P70 页 8。实验指导 p163 页 7。素数就是质数

算法描述：

判断 n 是否是质数，用 $2 \sim \sqrt{n}$ ($\text{sqrt}(n)$) 去除它，如果存在可以整除的情况，则 n 不是质数，否则必为质数。

```
#include <stdio.h>
```

```
#include <math.h>
```

```

int main( )
{
    int n, i, k;
    do
    {
        printf( "Enter a positive integer : " );
        scanf( "%d", &n );
    } while ( n <= 0 );                /* 确保 n 为正数 */
    if ( n == 1 )
    {
        printf( "%d is not a prime.\n", n );
    }
    else
    {
        k = (int)sqrt(n);
        for ( i = 2 ; i <= k ; i++ )
        {
            if ( n % i == 0 )
            {
                break;
            }
        }
        if ( i > k )
        {
            printf( "%d 是质数.\n", n );
        }
        else
        {
            printf( "%d 不是质数.\n", n );
        }
    }
}

```

```

    }
    return 0;
}

```

4 穷举法：百鸡百钱问题。p65 例 4.13 同类题目 P70 页 9。

问题描述:公鸡 5 钱 1 只, 母鸡 3 钱 1 只, 小鸡 1 钱 3 只。100 钱买 100 只鸡, 问公鸡、母鸡、小鸡各几只。

解决此问题的传统方式是设立方程组, 设公鸡、母鸡、小鸡的数量分别是 a 、 b 、 c , 则有

$$\begin{cases} a+b+c=100 \\ 5a+3b+c/3=100 \end{cases}$$

分析由于方程数量少于变量个数, 因此该方程组有多个解。

算法描述:

穷举法: 把解空间中的所有解都挨个尝试一遍, 判断是否符合问题的要求, 符合就作为求解结果。

```

#include <stdio.h>
int main()
{
    int a, b, c;
    for ( a = 0 ; a <= 20 ; a++ )
    {
        for ( b = 0 ; b <= 33 ; b++ )
        {
            for ( c = 0 ; c <= 100 ; c++ )
            {
                if ( a + b + c == 100 && 15 * a + 9 * b + c == 300 )
                {
                    printf( "%d, %d, %d\n", a, b, c );
                }
            }
        }
    }
    return 0;
}

```

当 a , b 确定时, c 也就确定了, 所以也可以用两层循环。代码如下。

```

#include <stdio.h>
int main()
{
    int a, b, c;
    for ( a = 0 ; a <= 20 ; a++ )
    {
        for ( b = 0 ; b <= 33 ; b++ )
        {
            c = 100 - a - b;
            if ( 15 * a + 9 * b + c == 300 )

```

```

        {
            printf( "%d, %d, %d\n", a, b, c );
        }
    }
}
return 0;
}

```

5 最大公约数:

算法: 用辗转相除法。具体见书 p70 页 5 题。

```

#include <stdio.h>
main()
{
    int a,b,d;
    scanf("%d%d",&a,&b);
    d=a%b;
    while(d!=0)
    {
        a=b;
        b=d;
        d=a%b;
    }
    printf("%d\n",b);
    return 0;
}

```

6 最小公倍数:

算法: 两个数的乘积除以最大公约数。

7 水仙花

C 语言算法总结 (7-11)

一、表达式

1 关于相等:

整型相等: a,b 为整型 `a==b`

double 类型相等: `fabs(a-b)<=1e-6`

字符串相等: `strcmp(a,b)==0`

2 关于赋值:

整型或 double 型变量之间赋值。 `a=b;`

字符串之间赋值 (字符串复制): `strcpy (a,b);`

3 字符串函数:

关于字符串, 用函数可以非常简洁方便, 掌握 `strcat`, `strcpy`, `strcmp` 的原型, `strcat`, `strcpy` 原理相同。

(1) `unsigned int strlen(const char *s);`

- (2) `char* strcpy(char *s1,const char *s2);`
- (3) `char* strcat(char *s1,const char *s2);`
- (4) `int strcmp(const char *s1,const char *s2);`
- (5) `char*strupr(char *s);`
- (6) `char* strlwr(char *s);`

4 指针

指针初始化: `int *p=&x;`或 `int *p,n;p=&x;`

指针赋值: 指针定义后要赋值, 如果不赋值, 则指针变量的值为随机值。

*和&互为逆运算

指针和一维数组: `int *p,x,a[5]={1,2,3,4,5};p=a;`

此时 `p` 指向 `a` 的首地址, 则有 `a[i],p[i],*(a+i),*(p+i)`等价。 `p` 也可以不指向 `a` 的首地址。

`a` 是数组名, `a` 里存放的是数组的首地址, `a` 也是一个指针常量, 可以有 `a+i`, 不可以有 `a++`, 因为 `a+i`, `a` 本身的值不发生改变, `a++`, `a` 本身的值会发生改变。

指针和二维数组: 利用指针找到具体的元素的地址, 有行地址和列地址之说。

行地址: `a,&a[0],a+i`, 每次+1, 移动 1 行, 会跳过数个元素。

列地址: `*a, *(a+i), a[0],&a[0][0]`, 每次+1, 移动 1 列。

举例: 若有定义 `inta[2][3]`, 对 `a[i][j]` 地址的正确引用有: `a[i]+j,(a+i)+j,*(a+i)+j`。请同学们自己区分哪个表达式是采用了行地址, 哪个表达式是采用了列地址的方法。

若有定义: `int *p=&x;` 请问 `*p++` 和 `(*p)++` 的区别?

二、程序片段

1 数组的逆序

2 选择排序

```
int    a;
int *p=&a;  相当于  int    a;    int *p;  p=&a;
int  a[];
int *p=a;  相当于  int  a[];  int *p;  p=a;
```

三经常出现的程序填空位置

注意: 红色部分为容易出填空的地方, 程序填空得分率低, 原因还是算法没有吃透。

1 `int search(const char *ps, const char *pf)`

```
{
    int count=0,i=0;
    char dest[20];          /*存储句子中的一个单词*/
    while(*ps)              /*判断字符串是否结束*/
    {
        i=0;
        while(((*ps>='a'&&*ps<='z')||(*ps>='A'&&*ps<='Z'))
        {
            dest[i++]=*ps++;
        }                  /*这个循环用于分词, 每个词存在数组 dest 中*/
        dest[i]='\0';
        ps++;               /*指向句子的下一个字符*/
        if(strcmp(dest,pf)==0) /*比较是否是待统计的单词*/
            count++;
    }
}
```



```

        return count;
    }
2 int main()
{
    char str[10];
    int i=0;
    printf("Input your password:\n");
    while( (str[i]=getchar()) != '#')          /* ' #' 作为结束标志 */
    {
        i++;
    }
    str[i]='\0';                               /* 增加字符串结束标志 */
    if (check(str))
        printf("Pass!\n");
    else
        printf("Error!\n\a\a\a");             /* 发出警报声 */
    return 0;
}
3 int main( )
{
    FILE *fp;
    char ch;
    fp = fopen( "C:\\dream.txt", "r+" );       /* 打开文本文件, 读写方式 */
    if ( fp == 0 )                             /* 文件打开失败 */
    {
        printf( "file error\n" );
        exit(1);
    }
    while( ( ch = fgetc(fp) ) != EOF)
    {
        if ( ch >= 'a' && ch <= 'z')
        {
            ch -= 32;
            fseek ( fp, -1L, 1 );               /* 回退 1 个字节 */
            fputc ( ch, fp );
            fflush( fp );                       /* 清空缓冲区, 将其中的内容输出 */
        }
    }
    fclose(fp);
    return 0;
}

```

4、以下程序功能是：读出 d1.c 的内容并输出至屏幕，请完善程序。

```

#include<stdio.h>
#include<stdlib.h>

```

```

int main()
{
    FILE *fp;
    char ch;
    if((fp=fopen("d1.c","r"))==NULL)
    {
        printf("Cannot open file! Please strike any key to exit!");
        getchar();
        exit(1);
    }
    ch=fgetc(fp);
    while (ch != EOF)
    {
        putchar(ch);
        ch=fgetc (fp);
    }
    fclose(fp);
    return 0;}

```

5、从键盘输入一段以“#”结尾的文本，将其输出至文本文件 D:\example.txt 中，同时统计其包含的大写字母数。

```

#include<stdio.h>
void main()
{
    FILE *fp;
    char ch;long count=0;
    if((fp=fopen("D:\\example.txt","w"))==NULL)
    {
        printf("Cannot open example.txt!\n");
        exit(0);
    }
    printf("input a string:\n");
    ch=getchar();
    while (ch!='#')
    {
        fputc(ch,fp);
        ch=getchar();
        if(ch>='A' && ch<='Z')
            count++;
    }
    printf("文件中大写字母个数为: %4d\n",count);
    fclose(fp);
}
printf("count=%d\n",count);
fclose(fp);
return 0;

```

```
}
```

6、已知有若干个学生的数据（包括学号、姓名和一门课程的成绩）。求课程的平均成绩，输出低于平均成绩的同学的学号、姓名及成绩。

```
#include <stdio.h>
struct student{
    int num;
    char name[8];
    int score;
};
int main()
{
    struct student stu[3]={{11,"tom",80},{12,"mike",90},{13,"helen",70}};
    int i;
    float ave=0;
    for(i=0;i<3;i++)
        ave=ave+ stu[i].score;
    ave=ave/3;
    for(i=0;i<3;i++)
        if( stu[i].score<ave)
            printf("%d,%s,%d\n",stu[i].num,stu[i].name,stu[i].score);
}
```

7、以下程序功能是：统计并输出文件 num.dat 中正整数、负整数、零的个数。（解析题）

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int positive=0,negative=0,zero=0,temp,x=0;
    FILE *fp;
    fp=fopen("num.dat","r");
    if(fp==NULL)
    {
        printf("Cannot open the file!\n");
        exit(0);
    }
    while(!feof(fp))
    {
        fscanf(______);
        if(temp>0) positive++;
        else if(temp<0) negative++;
        else ____;
    }
    ____;
```

```
printf("positive=%d,negative=%d,zero=%d\n",positive,negative,zero);  
return 0;  
}
```