

实训四 结构化程序设计与算法认识—— 循环结构程序设计

一、实训目的及要求

通过本次实训,掌握循环语句 for、while 和 do-while 的使用和 break 和 continue 的功能,并能熟练的进行应用。

程序 1、编程求 $1+2+3+\cdots+100$ 的和。

算法思路:

- 1、用变量 sum 作为累加器,存放和;
- 2、用变量 i 表示累加变量,分别存放 1,2,3,...,100;
- 3、当 $i \leq 100$ 时,执行 $\text{sum}=\text{sum}+i$;(等价于 $\text{sum}+=i$);
- 4、当 i 的值超过 100 的时候,不再执行 $\text{sum}=\text{sum}+i$; 输出 sum 的值。

方法一 (while 结构)

```
#include "stdio.h"
main()
{
    int sum=0, i=1;           /*变量初始化*/
    while (i<=100)
    {
        sum=sum+i;
        i++;
    }
    printf("sum is %d\n",sum);
}
```

方法二 (do-while 结构)

```
#include "stdio.h"
main()
{
    int sum=0, i=0;
    do
    {
        sum=sum+i;
        i++;
    } while (i<=100)
```

```
printf("sum is %d\n",sum);
}
```

方法三（for 结构）

```
#include "stdio.h"
{
    int i,sum=0;
    for(i=0;i<=100;i++)
        sum=sum+i;
    printf("sum=%d\n",sum);
}
```

程序 2、求 1 到 100 之间的奇数之和，偶数之积。

算法思路：

- 1、用变量 sum 存放奇数之和，用变量 mul 存放和；
- 2、用变量 i 表示累加变量,分别存放 1,2,3,...,100；
- 3、当 $i \leq 100$ 时，若 i 是奇数执行 $sum=sum+i$ ；
是偶数执行 $mul=mul*i$ ；
- 4、当 i 的值超过 100 的时候，不再执行求和和求积操作;输出 sum 和 mul 的值。

方法一（while 结构）

```
#include "stdio.h"
main()
{ double mul=1; /*双精度类型*/
  int i=1,sum=0;
  while (i<=100)
  {
      if (i%2!=0) sum=sum+i;
      else mul=mul*i;
      i++;
  }
  printf("sum=%d\n",sum);
  printf("mul=%e\n",mul); /*科学计数法输出*/
}
```

方法二（for 结构）

```
#include "stdio.h"
main()
{
    int i,sum=0;
    double mul=1;
```

```

for(i=1;i<=100;i++)
    if(i%2==1) sum=sum+i;
    else mul=mul*i;
printf("sum=%d,mul=%e\n",sum,mul);
}

```

程序 3、把输入的一行字符原样输出，若是大写字母需要转换成小写的。

算法思路：

- 1、通过键盘接收一个字符；
- 2、判断该字符是否为回车符；是转第 4 步，不是转到第 3 步；
- 3、判断接收的字符是否为大写字母，是转换为小写字母并输出，不是则直接输出，程序转回到第 1 步；
- 4、程序结束。

```

#include "stdio.h"
main()
{
    char ch;
    while((ch=getchar())!='\n')
    {
        if(ch>='A'&&ch<='Z')
            ch=ch+32;
        putchar(ch);
    }
}

```

程序 4、编写程序求 $2^0+2^1+2^2+2^3+\dots+2^{63}$ 的和。

算法思路：

变量设定：

- s: 存放累加和；
- t: 存放每项的值；
- i: 为循环次数；

算法分析：

- 1、循环累加，用 $s=s+t$ ； 或 $s+=t$ ；
- 2、每项可递推计算， $t=t*2$ ； 或 $t*=2$ ；
- 3、初值 $s=1,t=1$,循环 63 次。

```

#include "stdio.h"
main( )
{
    int i;
    float s=1,t=1;
    for (i=1;i<=63;i++)
    {
        t*=2;
        s+=t;
    }
    printf("s=%e\n",s);
}

```

程序 5、从键盘上任意输入一个正整数,判断其是否为素数。

素数：除了 1 和它本身之外没有其他因子的数。换句话说只要有因子（除了 1 和它本身之外）则该数一定不是素数

算法思路：

- 1、从键盘输入一正整数，存入变量 m 中；
- 2、测定 m 是否有约数，利用变量 $i=2-m-1$ ，逐个验证 $m\%i$ 是否等于 0；
- 3、如果 $m\%i==0$ ，说明 m 有约数，不符合素数的条件，后续的 i 不必验证，退出验证过程；
- 4、根据退出的条件判定 m 是否有约数，若非正常退出说明 m 有约数 i ，否则没有约数，即为素数。

```

#include "stdio.h"
main( )
{
    int m,i;
    scanf("%d",&m);
    for(i=2;i<=m-1;i++)
        if(m%i==0) break;
    if(i>m-1) printf("%d is a prime",m);
    else printf("%d is not a prime",m);
}

```

实训题目

- 1、任意输入两个整数,求它们的最大公约数。
- 2、打印形状为直角三角形的乘法口诀。
- 3、输入一个包含有三个数字字符的字符串，将该字符串转换为整形数据输

出。（如：输入字符串 123，则输出整形数据 123）