**实验三 循环程序设计实验**

**1.实验目的**

1.掌握循环程序的设计方法。

2.掌握比较指令、转移指令和循环指令的使用方法。

3.进一步掌握调试工具的使用方法。

**2.实验内容**

**题目一**

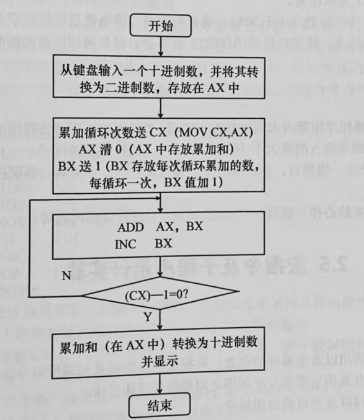
**1.实验内容**

计算1+2+..-n=?，其中n通过键盘输入，累加和小于65535。要求在屏幕上提供如下信息:

Please input a number(1-361): 出现此信息后通过键盘输入一个小于362的无符号整数;使累加和小于一个16位无符号二进制数所能表示的范围。

1+2+...+n=sum ;其中n为用户输入的数，sum为所求的累加和。

**2.程序流程图**



**3.程序代码**

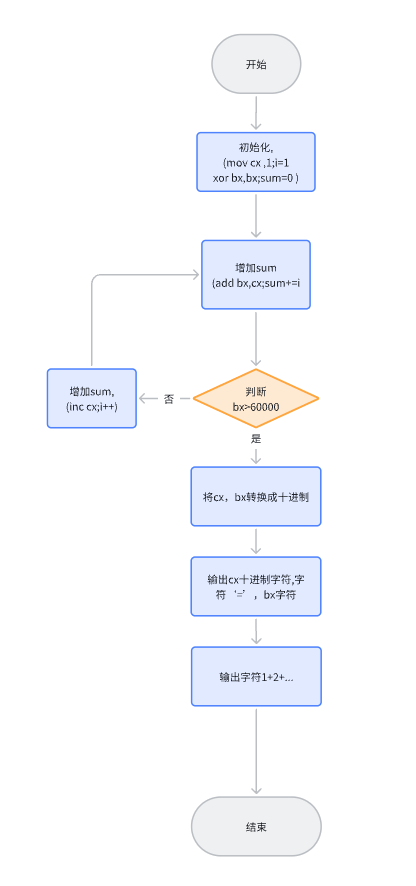
|  |
| --- |
| XML  STACK SEGMENT STACK  DB 100 DUP(?) STACK ENDS  DATA SEGMENT  INF1 DB "Please input a number N(0-627): $"  IBUF DB 7,0,6 DUP(?)  OBUF DB 6 DUP(?)  INF2 DB 0AH,0DH,"$" DATA ENDS  CODE SEGMENT  ASSUME CS:CODE,DS:DATA  START:  MOV AX,DATA  MOV DS,AX  DSTRING MACRO STRING  PUSH DX  PUSH AX  MOV DX,OFFSET STRING  MOV AH,09H  INT 21H  POP AX  POP DX ENDM   DSTRING INF1  CALL DTOB  CALL ADDN  CALL BTOAD  MOV AH,4CH  INT 21H  DTOB PROC NEAR  PUSH CX  MOV DX,OFFSET IBUF  MOV AH,0AH  INT 21H  MOV CL,IBUF+1  MOV CH,0  MOV SI,OFFSET IBUF+2  MOV AX,0  AGAIN:  MOV DX,10  MUL DX  AND BYTE PTR [SI],0FH  ADD AL,[SI]  ADC AH,0  INC SI  LOOP AGAIN   POP CX  RET DTOB ENDP  ADDN PROC NEAR  MOV CX,AX  XOR AX,AX  MOV BX,01H  LOOP5:  ADD AX,BX  INC BX  LOOP LOOP5  RET ADDN ENDP  BTOAD PROC NEAR  MOV BX,OFFSET OBUF+5  MOV BYTE PTR [BX],'$'  MOV CX,10  LOOP4:  MOV DX,0  DIV CX  ADD DL,30H  DEC BX  MOV [BX],DL  XOR AX,0  JNZ LOOP4   MOV DX,OFFSET INF2  MOV AH,09H  INT 21H   MOV DX,BX  MOV AH,09H  INT 21H  RET BTOAD ENDP  CODE ENDS END START |

**2.2 题目二**

**1.实验内容**

从自然数1开始累加，直到累加和大于60000为止，显示累加的自然数的个数和累加和。显示1+2+...+n=sum，其中n为累加个数，sum 为累加和。

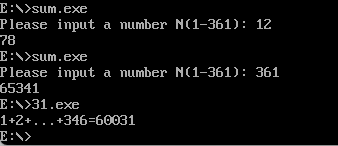
**2.程序流程图**

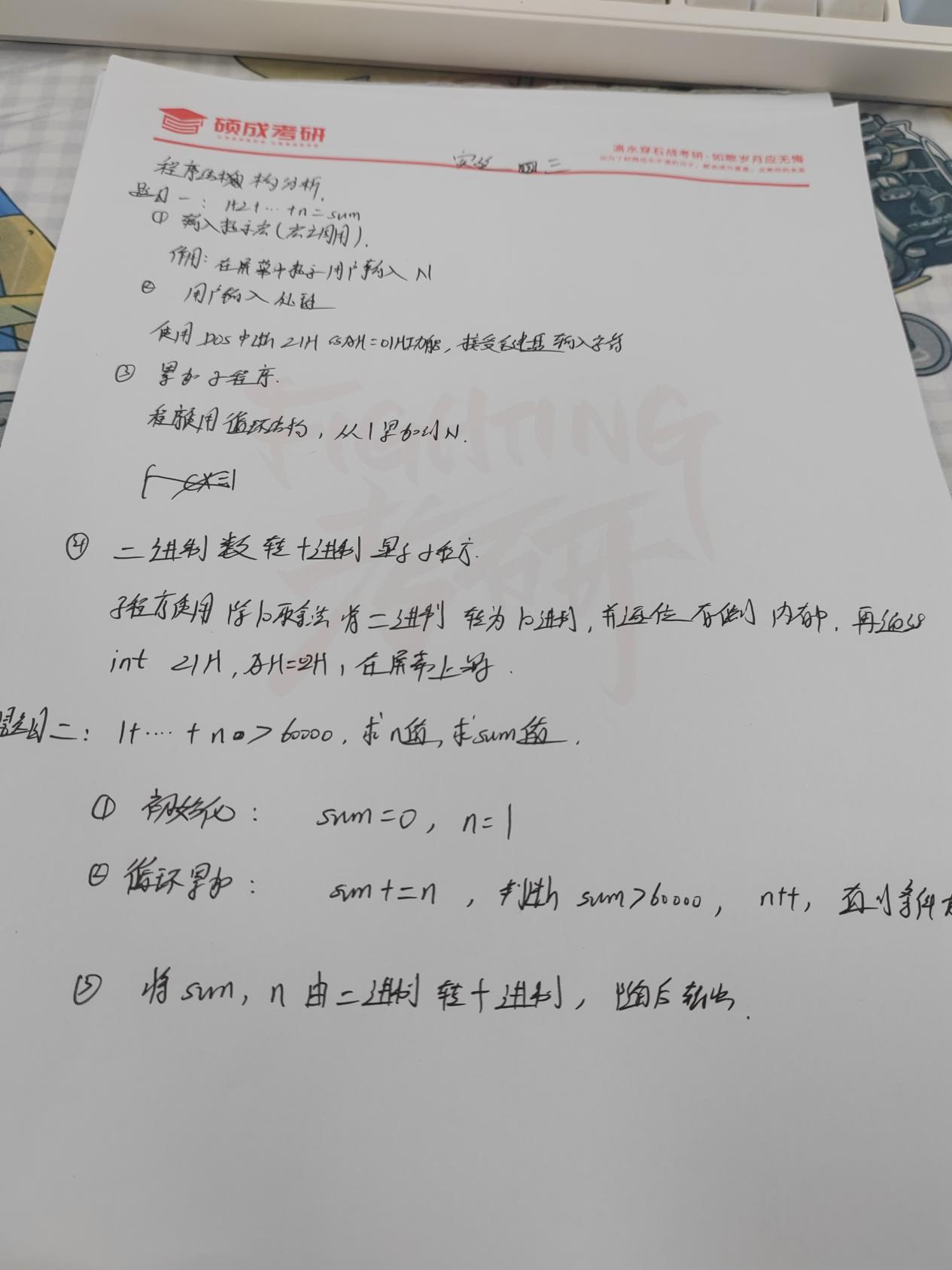


**3.程序代码**

|  |
| --- |
| XML DATA SEGMENT  msg1 db '1+2+...+', '$'  msg2 db '=', '$'  numStr db 6 dup(0)  sumStr db 6 dup(0) DATA ENDS start:  mov ax, @data  mov ds, ax   ; 初始化  mov cx, 1 ;   xor bx, bx ;  sum\_loop:  add bx, cx ;   cmp bx, 60000  ja output   inc cx ;  jmp sum\_loop  output:    ; 输出 "1+2+...+"  mov dx, offset msg1  call printStr   ; 输出 n  push bx;  mov ax, cx  mov si, offset numStr  call printNum  mov dx, si  call printStr   ; 输出 "="  mov dx, offset msg2  call printStr   ; 输出 sum  pop bx;  mov ax, bx  mov si, offset sumStr  call printNum  mov dx, si  call printStr   ; 退出程序  mov ah, 4ch  int 21h  ; 将 AX 转为字符串（以 '$' 结尾），结果保存在 SI 指向的缓冲区 printNum:  mov bx, 10  mov di, si  add di, 5  mov byte ptr [di], '$'  dec di nextDigit:  xor dx, dx  div bx  add dl, '0'  mov [di], dl  dec di  cmp ax, 0  jne nextDigit  inc di  mov si, di  ret  ; 输出以 '$' 结束的字符串（DX 指向） printStr:  mov ah, 09h  int 21h  ret  end start |

**3.实验结果**





**4.实验总结**

