



哈尔滨工业大学（深圳）

实践教学报告

学 院： 计算机科学与技术学院

题 目： 实验三：子程序和宏设计实例

姓 名： 李秋阳

学 号： 180110527

专 业： 计算机科学与技术

日 期： 2019 年 10 月 24 日

一、问题描述

必做题：

题目：从屏幕输入一个十进制正数（0 到 2000 之间），将其转换成 K 进制（K=2、4、8、16）并将其转换结果在屏幕上输出出来。

对输入的正数进行处理，然后根据所给的进制求出每一位，逐个输出

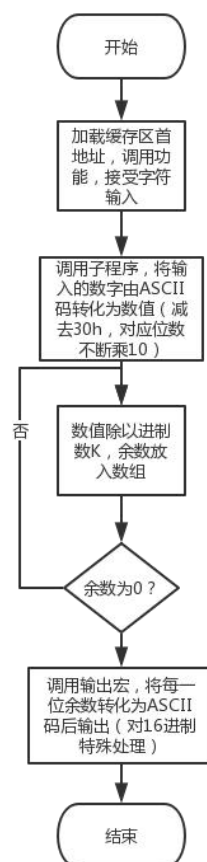
选做题 1：

题目：求斐波那契数，程序接受用户输入的 n 值，n 值范围为 1-20。根据给定 n 值，计算斐波那契数

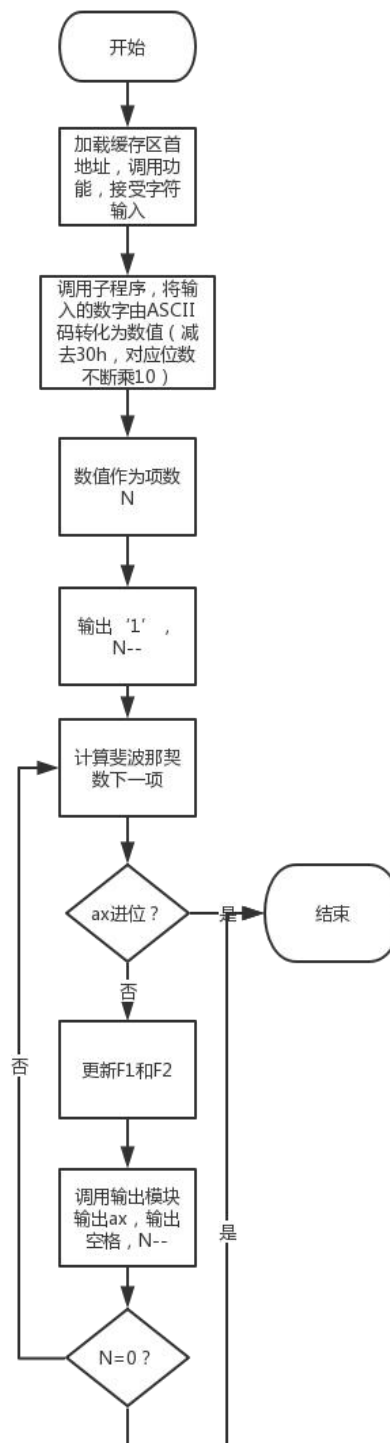
设计递归程序，实现对斐波那契数的求解

二、解决方案

必做题：调用功能接受一个数字的输入，设计子程序，通过循环，将输入的十进制数字的 ASCII 码转化为可供处理的数值。对转化后的数字通过除数取余法找到对应进制转化后各位数字的数值，覆盖原有数组。对得到的已经成功转化的数字调用宏定义的 print 功能，先根据需求转化为数字的 ASCII 码以保证输出正确。



选做题 1：调用功能接受一个数字的输入，设计子程序，通过循环，将输入的十进制数字的 ASCII 码转化为可供处理的数值。先对不符合递归规律的第一和第二项直接赋值，然后调用子程序，通过循环实现对斐波那契数的递归求解 ($FIB(n) = FIB(n-2) + FIB(n-1)$ ($n > 2$))，输出得到的值，输出空格，继续循环，直到完成对整个斐波那契数列的求解。



三、具体实现

必做题

```

;将栈中字符取出并输出
print macro x
    mov cx,x
pri:
    pop ax
    cmp ah,10
    jb next
    add ah,7h
next:
    add ah,30h
    mov dl,ah
    mov ah,02h
    int 21h

    loop pri
endm

datas segment
    buff db 6 ;定义缓存区的大小
    db ?
    db 20 dup(?)
    input dw 0
    k db 16
datas ends

stacks segment
;此处输入堆栈段代码
stacks ends

codes segment
    assume cs:codes,ds:datas,ss:stacks
start:
    mov ax,datas
    mov ds,ax
    lea dx,buff ;加载缓存区首地址
    mov ah,0ah ;功能0ah送ah
    int 21h
    mov cl,buff[1] ;获取实际数的位数
    xor ch,ch
    mov si,2
    call get10
    mov ax,di
    mov bx,0

tok:;通过不断除k的方式得到k进制的各个位
    div k
    push ax
    mov ah,0
    inc bx
    cmp ax,0
    jnz tok
    print bx
    mov ah,4ch
    int 21h

get10 proc near;将字符串转成数字
getit:
    mov bl,buff[si]
    sub bl,30h
    mov bh,0
    push bx
    inc si
    loop getit
    mov cl,buff[1]
    mov di,0
    mov bx,1
to10:
    mov ax,0
    pop ax
    mul bx
    add di,ax
    mov ax,bx
    mov bx,10
    mul bx
    mov bx,ax
    loop to10
    ret
get10 endp
codes ends
end start
```

结果

```
D:\>trans
789 ▲
```

```
D:\>trans
315
```

```

D:\>debug trans.exe
-a
076C:0000
-u
076C:0000 B86A07      MOV     AX,076A
076C:0003 8ED8          MOV     DS,AX
076C:0005 8D160000      LEA     DX,[0000]
076C:0009 B40A          MOV     AH,0A
076C:000B CD21          INT     21
076C:000D 8A0E0100      MOV     CL,[0001]
076C:0011 32ED          XOR     CH,CH
076C:0013 BE0200      MOV     SI,0002
076C:0016 E82C00      CALL    0045
076C:0019 8BC7          MOV     AX,DI
076C:001B BB0000      MOV     BX,0000
076C:001E F6361800      DIV     BYTE PTR [0018]
-r dx
DX 0000
:0789

```

```

-g41
315
AX=0235 BX=0003 CX=0000 DX=0035 SP=0000 BP=0000 SI=0005 DI=0315
DS=076A ES=075A SS=0769 CS=076C IP=0041 NU UP EI PL NZ NA PE NC
076C:0041 B44C          MOV     AH,4C

```

选做题一

```

assume cs:code,ds:data
data segment
    msg1 db 13,10,'please input the num of fibonacci, no larger
than 256, n= $'
    msg2 db 13,10,'fibonacci sequence is: $'    ;提示信息

    n dw 0
    f1 dw 0
    f2 dw 1
data ends

code segment
start:
    mov ax,data
    mov ds,ax
    lea dx,msg1
    mov ah,9
    int 21h
    call input
    cmp cx,1
    jb exit
    mov n,cx

    lea dx,msg2
    mov ah,9
    int 21h
    mov dl,'1'
    mov ah,2
    int 21h
    mov dl,' '
    int 21h
    dec n
    jz exit

loop0:
    mov ax,f1
    add ax,f2
    jc exit
    mov bx,f2
    mov f1,bx
    mov f2,ax
    call output
    mov dl,' '
    mov ah,2
    int 21h
    dec n
    jnz loop0

exit:
    mov ah,4ch
    int 21h

```

```

input:
    mov bl,10
    mov cx,0

in_x:

    mov ah,7
    int 21h

    cmp al,13
    je in_end

    cmp al,'0'
    jb in_x
    cmp al,'9'
    ja in_x

    mov dl,al

    mov ah,2
    int 21h
    mov al,dl
    sub al,30h
    mov ah,0
    xchg ax,cx
    mul bl
    add cx,ax
    cmp ch,0
    jnz in_end
    jmp in_x

in_end:
    ret
output:
    mov bx,10
    mov cx,0

loop1:
    mov dx,0
    div bx
    add dl,'0'
    push dx
    inc cx
    cmp ax,0
    jnz loop1
    mov ah,2
loop2:
    pop dx
    int 21h
    loop loop2
    ret
code ends
end start

```

结果

```

D:\>feibo

please input the num of fibonacci,no lager than 256,n= 19
fibonacci sequence is: 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584
4181

```

```

-u
0770:0000 B86A07      MOV     AX,076A
0770:0003 8ED8        MOV     DS,AX
0770:0005 8D160000      LEA     DX,[0000]
0770:0009 B409        MOV     AH,09
0770:000B CD21        INT     21
0770:000D E84800      CALL    0058
0770:0010 83F901      CMP     CX,+01
0770:0013 723F        JB      0054
0770:0015 890E5400     MOV     [0054],CX
0770:0019 8D163A00     LEA     DX,[003A]
0770:001D B409        MOV     AH,09
0770:001F CD21        INT     21

```

```

-u
0770:0021 B231        MOV     DL,31
0770:0023 B402        MOV     AH,02
0770:0025 CD21        INT     21
0770:0027 B220        MOV     DL,20
0770:0029 CD21        INT     21
0770:002B FF0E5400     DEC     WORD PTR [0054]
0770:002F 7423        JZ      0054
0770:0031 A15600      MOV     AX,[0056]
0770:0034 03065800     ADD     AX,[0058]
0770:0038 721A        JB      0054
0770:003A 8B1E5800     MOV     BX,[0058]
0770:003E 891E5600     MOV     [0056],BX

```

```

-u
0770:0042 A35800      MOV     [0058],AX
0770:0045 E83E00      CALL    0086
0770:0048 B220        MOV     DL,20
0770:004A B402        MOV     AH,02
0770:004C CD21        INT     21
0770:004E FF0E5400     DEC     WORD PTR [0054]
0770:0052 75DD        JNZ     0031
0770:0054 B44C        MOV     AH,4C
0770:0056 CD21        INT     21
0770:0058 B30A        MOV     BL,0A
0770:005A B90000      MOV     CX,0000
0770:005D B407        MOV     AH,07
0770:005F CD21        INT     21
0770:0061 3C0D        CMP     AL,0D

```

```

-g54
please input the num of fibonacci, no larger than 256, n= 19
fibonacci sequence is: 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584
4181
AX=0220 BX=000A CX=0000 DX=0020 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=0770 IP=0054 NU UP EI PL ZR NA PE NC
0770:0054 B44C      MOV     AH,4C
-d0
076A:0000 0D 0A 70 6C 65 61 73 65-20 69 6E 70 75 74 20 74  ..please input t
076A:0010 68 65 20 6E 75 6D 20 6F-66 20 66 69 62 6F 6E 61  he num of fibona
076A:0020 63 63 69 2C 6E 6F 20 6C-61 67 65 72 20 74 68 61  cci, no larger tha
076A:0030 6E 20 32 35 36 2C 6E 3D-20 24 0D 0A 66 69 62 6F  n 256, n= $.fibo
076A:0040 6E 61 63 63 69 20 73 65-71 75 65 6E 63 65 20 69  nacci sequence i
076A:0050 73 3A 20 24 00 00 18 0A-55 10 00 00 00 00 00 00  s: $....U.....
076A:0060 B8 6A 07 8E D8 8D 16 00-00 B4 09 CD 21 E8 48 00  .j.....!H.
076A:0070 83 F9 01 72 3F 89 0E 54-00 8D 16 3A 00 B4 09 CD  ...r?..T...:....

```

四、总结

在本次实验中，我们主要学习了通过子程序和宏设计使得汇编语言编程更有效率和条理性的操作。如何将相似的功能函数整合到一起，减少代码工程量，便于调试，是这次实验中我们所要思考的问题。本次实验大大提高了我对子程序设计和高级语言汇编技术的理解，受益匪浅。