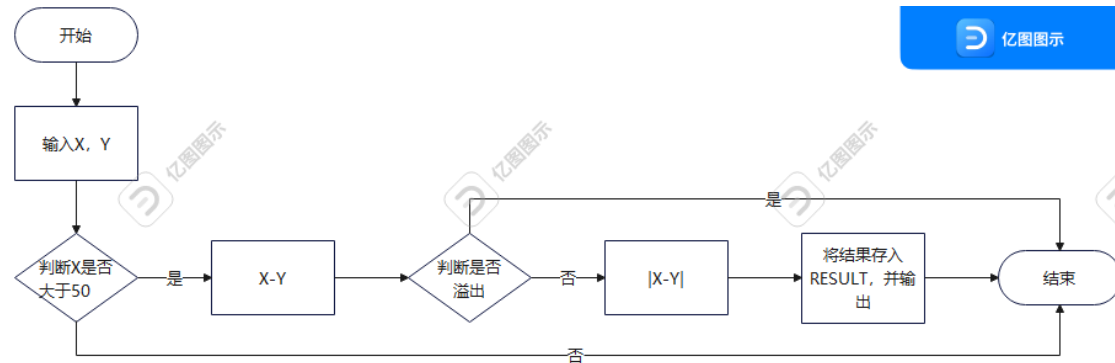


1. 设 X、Y 均为存放在 X 和 Y 单元中的 16 位操作数，先判 $X > 50$ 否，如不满足，转到 T,如满足条件则转移到 TOO HIGH 去执行,然后故 $X - Y$,如溢出则转移到 OVERFLOW 去执行,否则计算 $X - Y$,并把结果存入 RESULT 中。画出程序流程图，并给出程序代码。



2. 代码:

```
assume ds:data,ss:stacks,cs:code
```

```
data segment
```

```
    RESULT db 4 dup(0)
```

```
    temp0 db "Please input numder X:$"
```

```
    temp1 db "Please input number Y:$"
```

```
    CRLF db 0ah,0dh,"$"
```

```
data ends
```

```
stacks segment stack
```

```
    dw 10 dup(0)
```

```
stacks ends
```

```
code segment
```

```
start:
```

```
    mov ax,data
```

```
    mov ds,ax
```

```
    mov ax,stacks
```

```
    mov ss,ax
```

```
    lea dx,temp0
```

```
    mov ah,09h
```

```
    int 21H
```

```
    call input;将输入的多位数字放入 ax 寄存器中,此处输入 X 值
```

```
    push ax
```

```
    lea dx,temp1
```

```
    mov ah,09h
```

```
    int 21H
```

```
    call input;输入 Y 值
```

```
    pop bx
```

```
    cmp bx,50
```

```
    jna EXIT
```

```

    cmp bx,ax;比较 X 和 Y 的大小，防止溢出
    jo EXIT;如果溢出就退出
    cmp bx,ax
    jb S0;如果 X<Y
    sub bx,ax
    mov ax,bx
    call output
    jmp EXIT
S0:
    sub ax,bx
    push ax
    mov dl,'-'
    mov ah,02h
    int 21H
    pop ax
    call output;将 ax 寄存器中的多位数字输出

EXIT:
    mov ax,4c00h
    int 21H

;input
input proc near
    mov cl,10
    mov bx,0
loop0:
    mov ah,1
    int 21h
    cmp al,'0'
    jb over
    cmp al,'9'
    ja over
    sub al,48
    mov dl,al
    mov ax,bx
    mul cl
    mov dh,0;这一步已经将十进制数字转化为 16 进制数字，所以说可以直接使用
sub,add
    add ax,dx
    mov bx,ax
    jmp loop0
over:
    mov ax,bx
    ret

```

```

input endp
;output
output proc near
    mov bl,0
    mov cl,10
loop1:
    div cl
    push ax;直接将结果的商和余数全部压栈
    add bl,1
    mov ah,0
    cmp ax,0
    jne loop1

    ;将所有的数据弹栈并输出
loop2:
    pop dx
    mov dl,dh;将余数移到输出位置
    sub bl,1
    add dl,48
    mov ah,2
    int 21h
    cmp bl,0
    jne loop2

    ret
output endp

code ends
end start

```