

## PHẦN II:

### Câu 1:

Vẽ lưu đồ và viết chương trình con TINHTB tính giá trị trung bình của 10 số BCD (dạng không nén) chứa trong các ô nhớ ram nội từ địa chỉ 30h. kết quả (phần nguyên) chứa trong thanh ghi R7.

```
tinhtungbinh:
    mov r0,#30h
    mov r1,#10
cong:
    add a,@r0
    da a
    inc r0
    djnz r1,cong
    mov b,#10h
    div ab
    mov r7,a
    ret
```

**Câu 2:** Vẽ lưu đồ và viết chương trình cộng 10 số nguyên không dấu chứa trong ram nội từ địa chỉ 30h. kết quả đặt trong R6 (byte thấp) , R7 (byte cao).

```
org 00h
    mov r0,#30h
    mov r1,#10h
loop:
    add a,@r0
    jnc No_cy
    inc r7
No_cy:
    inc r0
    djnz r1,loop
    mov r6,a
end
```

**3. Số A6h được đặt trong 1 ô nhớ ram ngoài trong khoảng địa chỉ từ 0100h đến 0150h. Vẽ lưu đồ và viết chương trình tìm địa chỉ của ô nhớ này rồi đặt trong R7 (byte cao) và R6 (byte thấp).**

```
org 00h
    mov dptr,#100h
    mov r1,#51
loop:
    clr a
    movc a,@a+dptr
    cjne a,#0a6h,tieptuc
    sjmp laydiachi
tieptuc:
    inc dptr
    djnz r1,loop
    sjmp exit
laydiachi:
    mov a,dpl
    mov r6,a
    mov a,dph
    mov r7,a
exit:
    nop
end
```

**4. Vẽ lưu đồ và viết chương trình con xét giá trị của chân P1.0 và lấy bù số 16bit cất trong 2 thanh ghi R7\_R6 (R7: byte cao, R6: byte thấp). Nếu P1.0 = 0 thì lấy bù 1, P1.0 = 2 thì lấy bù 2, kết quả trả về cất trong R7\_R6.**

```
abc:
    jnb P1.0,bu1
bu2:
    acall bu1
    mov a,r6
    add a,#1
    jnc next
    inc r7
    ret
next:
    mov r6,a
    ret
bu1:
    mov a,r7
    cpl a
    mov r7,a
```

```

mov r6,a
cpl a
mov r6,a
ret

```

**5. Vẽ lưu đồ và viết chương trình tìm giá trị lớn nhất trong 20 số không dấu 8 bit được đặt trong Ram nội từ địa chỉ 30h. kết quả đặt trong thanh ghi R7.**

```

org 00h
    mov r0,#30h
    mov r1,#19
    mov a,@r0

main:
    inc r0
    mov b,@r0
    cjne a,b,kiemtra
    sjmp tieptuc
kiemtra:
    jnc tieptuc
    mov a,b
tieptuc:
    djnz r1,main
    mov r7,a
end

```

**6. Cho mạch kết nối như sau:**

**Hãy vẽ lưu đồ và viết chương trình để sáng từng LED theo chiều D8 ® D1 và lặp lại, thời gian sáng của mỗi LED là 0,4s. Giả sử hệ thống dùng thạch anh 6MHz.**

```

ORG 00H
NGHICH: MOV P0,#11111111B
        CALL DELAY
        CLR  P0.7
        CALL DELAY
        SETB P0.7
        CLR  P0.6

```

```

CALL DELAY
SETB P0.6
CLR  P0.5

```

```

CALL DELAY
SETB P0.5
CLR  P0.4
CALL DELAY
SETB P0.4
CLR  P0.3
CALL DELAY
SETB P0.3

```

```

    CLR  P0.2
CALL DELAY
SETB P0.2
CLR  P0.1
CALL DELAY
SETB P0.1
CLR  P0.0
CALL DELAY
SETB P0.0
RET

```

```

DELAY:  MOV R1,#2
LAP1:  MOV R2,#250
LAP2:  MOV R3,#200
LAP3:  DJNZ R3,$
        DJNZ R2,LAP2
        DJNZ R1,LAP1
RET

```

END

**7. Cho mạch kết nối như sau:**

**Hãy vẽ lưu đồ và viết chương trình để sáng từng LED theo chiều D1 ® D8 rồi**

**D8 ® D1 và lặp lại, thời gian sáng của mỗi LED là 0,3s. Giả sử hệ thống dùng thạch anh 12MHz.**

ORG 00H

MAIN:        LCALL THUAN  
              LCALL DELAY  
              LCALL NGHICH  
              LJMP MAIN

THUAN:      MOV P0,#11111111B  
              CALL DELAY  
              CLR    P0.0  
              CALL DELAY  
              SETB P0.0  
              CLR    P0.1  
              CALL DELAY  
              SETB P0.1  
              CLR    P0.2  
              CALL DELAY  
              SETB P0.2  
              CLR    P0.3  
              CALL DELAY  
              SETB P0.3  
              CLR    P0.4  
              CALL DELAY  
              SETB P0.4  
              CLR    P0.5  
              CALL DELAY  
              SETB P0.5  
              CLR    P0.6  
              CALL DELAY  
              SETB P0.6  
              CLR    P0.7

CALL DELAY

SETB P0.7

RET

NGHICH: MOV P0,#11111111B

CALL DELAY

CLR    P0.7

CALL DELAY

SETB P0.7

CLR    P0.6

CALL DELAY

SETB P0.6

CLR    P0.5

CALL DELAY

SETB P0.5

CLR    P0.4

CALL DELAY

SETB P0.4

CLR    P0.3

CALL DELAY

SETB P0.3

CLR    P0.2

CALL DELAY

SETB P0.2

CLR    P0.1

CALL DELAY

SETB P0.1

CLR    P0.0

CALL DELAY

SETB P0.0

RET

DELAY:      MOV R1,#20

```

LAP1: MOV R2,#81
LAP2: MOV R3,#86
LAP3: DJNZ R3,LAP3
      DJNZ R2,LAP2
      DJNZ R1,LAP1
RET

```

END

**8. Vẽ lưu đồ và viết chương trình chuyển số BCD(không nén) trong hai thanh ghi R7(byte cao) và R6(byte thấp) thành số HEX và cất trong ô nhớ Ram ngoài có địa chỉ là 1000h.**

```

org 00h
    sjmp main

main:
    mov a,r7
    swap a
    orl a,r6
    mov dptr,#1000h
    movx @dptr,a
end

```

**10. Cho mạch kết nối như sau:**

**Hãy vẽ lưu đồ và viết chương trình để hiển thị giá trị BCD trong ô nhớ 20H lên LED 7 đoạn, 1s sau thì hiển thị giá trị BCD trong ô nhớ 21H lên LED 7 đoạn, 1s sau thì lặp lại. Biết  $f_{osc} = 12\text{MHz}$ . Giả sử có sẵn bảng mã LED 7 đoạn trong bộ nhớ chương trình tại địa chỉ tương ứng với nhãn TABLE (không cần lập bảng mã).**

```

ORG 00H
MOV 20H, #2
MOV 21H, #6
MAIN: CALL LOOP1
      CALL DELAY

```

```

CALL LOOP2
CALL DELAY
JMP MAIN

```

```

LOOP1: MOV A,20H
      MOV DPTR, #TABLE
      MOVC A,@A+DPTR
      MOV P0,A
      RET

```

```

LOOP2: MOV A, 21H
      MOV DPTR, #TABLE
      MOVC A,@A+DPTR
      MOV P0,A
      RET

```

TABLE:

```

DB 0C0H, 0F9H, 0A4H, 0B0H,
   99H, 92H, 82H, 0F8H, 80H, 90H

```

```

DELAY: MOV R1,#79
      LAP1: MOV R2,#80
      LAP2: MOV R3,#77
      LAP3: DJNZ R3, LAP3
            DJNZ R2, LAP2
            DJNZ R1, LAP1
            RET

```

END

**11. Cho mạch kết nối như sau:**

**Hãy vẽ lưu đồ và viết chương trình để các LED sáng lan tỏa theo chiều D1 ® D8 rồi tắt dần theo chiều D1 ® D8 và lặp lại, thời gian cách nhau giữa 2 LED là 0,4s. Giả sử hệ thống dùng đồng thạch anh 12MHz.**

```

ORG 00H

```

```

MAIN:      LCALL THUAN
           LCALL DELAY
           LCALL NGHICH
           LCALL DELAY
           LJMP MAIN
THUAN:     MOV  P0,#11111111B
           CALL DELAY
           CLR   P0.0
           CALL DELAY
           CLR   P0.1
           CALL DELAY
           CLR   P0.2
CALL DELAY
           CLR   P0.3
           CALL DELAY
           CLR   P0.4
           CALL DELAY
           CLR   P0.5
           CALL DELAY
           CLR   P0.6
           CALL DELAY
           CLR   P0.7
           CALL DELAY

           RET
NGHICH:    SETB P0.0
           CALL DELAY
           SETB P0.1
           CALL DELAY
           SETB P0.2
           CALL DELAY
           SETB P0.3

```

```

           CALL DELAY
           SETB P0.4
           CALL DELAY
           SETB P0.5
           CALL DELAY
           SETB P0.6
           CALL DELAY
           SETB P0.7
           CALL DELAY
           RET
DELAY:     MOV  R1,#20
           LAP1: MOV R2,#81
           LAP2: MOV R3,#86
           LAP3: DJNZ R3,LAP3
                DJNZ R2,LAP2
                DJNZ R1,LAP1
           RET
END

```

## 12. Cho mạch kết nối như sau:

**Hãy vẽ lưu đồ và viết chương trình để hiển thị giá trị BCD trong ô nhớ 30H lên LED 7 đoạn, 0.5s sau thì hiển thị giá trị BCD trong ô nhớ 31H lên LED 7 đoạn, 0.5s sau thì lặp lại. Giả sử hệ thống dùng thạch anh 12MHz. Giả sử có sẵn bảng mã LED 7 đoạn trong bộ nhớ chương trình tại địa chỉ tương ứng với nhãn TABLE.**

```

ORG 00H
MOV 30H, #2
MOV 31H, #6
MAIN: CALL LOOP1
      CALL DELAY

```

```

CALL LOOP2
CALL DELAY
JMP MAIN
LOOP1: MOV A,30H
      MOV DPTR, #TABLE
      MOVC A,@A+DPTR
      MOV P0,A
      RET
LOOP2: MOV A, 31H
      MOV DPTR,#TABLE
      MOVC A,@A+DPTR
      MOV P0,A
      RET
TABLE: DB 0C0H, 0F9H, 0A4H, 0B0H, 99H,
92H, 82H, 0F8H, 80H, 90H
DELAY: MOV R1,#30
      LAP1: MOV R2,#90
      LAP2: MOV R3,#88
      LAP3: DJNZ R3, LAP3
            DJNZ R2, LAP2
            DJNZ R1, LAP1
      RET

```

**13. Hãy vẽ lưu đồ và viết chương trình tìm giá trị lớn nhất trong các số không dấu được đặt trong các thanh ghi R0 , R4, kết quả đặt trong thanh ghi R7.**

```

ORG 0000H
JMP MAIN
MAIN:
MOV A,R0
MOV B,R1
CJNE A,B,$+3
JNC LHB
JC NHO
LHB:

```

```

MOV R7,A
JMP X1
NHO:
MOV R7,B
JMP X1
X1:
MOV A,R7
MOV B,R2
CJNE A,B,$+3
JNC LHB1
JC NHO1
LHB1:
MOV R7,A
JMP X2
NHO1:
MOV R7,B
JMP X2
X2:
MOV A,R7
MOV B,R3
CJNE A,B,$+3
JNC LHB2
JC NHO2
LHB2:
MOV R7,A
JMP X3
NHO2:
MOV R7,B
JMP X3
X3:
MOV A,R7
MOV B,R4
CJNE A,B,$+3
JNC LHB3
JC NHO3
LHB3:
MOV R7,A
RET
NHO3:
MOV R7,B
RET

```

**Cách 2:**

```

org 00h
    sjmp main
main:
    mov 32h,r2
    mov 33h,r3
    mov 34h,r4

    mov a,r0
    mov b,r1
    mov 40h,r0
    mov r0,#31h
loop:
    cjne a,b,kiemtra
    sjmp tieptuc
kiemtra:
    jnc tieptuc
    mov a,b
tieptuc:
    inc r0
    mov b,@r0
    cjne r0,#35h,loop
    mov r7,a
    mov r0,40h
end

```

**Câu 14:**  
**Vẽ lưu đồ và viết chương trình con bintobcd chuyển số nhị phân trong A sang số BCD 3 digit cất trong các ô nhớ 32h: hàng trăm, 31h: hàng chục, 30h: hàng đơn vị.**

```

bintobcd:
    mov b,#10
    div ab
    mov 30h,b
    cjne a,#10,kiemtra
    sjmp chiatiep
kiemtra:
    jc khongchianua
chiatiep:
    mov b,#10
    div ab
    mov 31h,b
    mov 32h,a

```

```

ret
khongchianua:
    mov 31h,a
    mov 32h,#0
    ret
end

```

**15. Vẽ lưu đồ và viết chương trình con bcdtobin chuyển số BCD nén (2 ký số) trong thanh ghi A thành số nhị phân cất trong ô nhớ 30h.**

```

bcdtobin:
    mov r2,a
    anl a,#0f0h
    swap a
    mov b,#10
    mul ab
    mov r0,a
    mov a,r2
    anl a,#0fh
    add a,r0
    mov 30h,a
    ret
end

```

**16. Vẽ lưu đồ và viết chương trình tìm giá trị nhỏ nhất trong các số không dấu được đặt trong các thanh ghi R0 , R4, kết quả cất vào ô nhớ 0030H trong RAM ngoài.**

```

org 00h
    sjmp main
main:
    mov 32h,r2
    mov 33h,r3
    mov 34h,r4

    mov a,r0
    mov b,r1
    mov 40h,r0
    mov r0,#31h
loop:
    cjne a,b,kiemtra

```

```

        sjmp tieptuc
kiemtra:
        jc tieptuc
        mov a,b
tieptuc:
        inc r0
        mov b,@r0
        cjne r0,#35h,loop
        mov r0,40h
        mov dptr,#30h
        movx @dptr,a
end

```

**17. Vẽ lưu đồ và viết chương trình con đổi các ký tự chữ hoa trong các ô nhớ 20h - 2Fh trong Ram nội sang chữ thường và ngược lại, rồi cất vào trong các ô nhớ Ram ngoài có địa chỉ 8400h - 840Fh.**

```

hoasangthuongnguoclai:
        mov dptr,#8400h
        mov r0,#20h
        acall loop
        ret
loop:
        mov a,@r0
        cjne a,#61h,next
        sjmp chuthuong
next:
        jc chuhoa
        sjmp chuthuong
chuhoa:
        add a,#20h
        movx @dptr,a
        inc r0
        inc dptr
        sjmp next2
chuthuong:
        anl a,#0dfh
        movx @dptr,a
        inc r0
        inc dptr
next2:

```

```

        cjne r0,#30h,loop
        ret

```

**19. vẽ lưu đồ và viết chương trình đổi số HEX chứa trong ô nhớ 30h thành mã ascii tương ứng chứa trong hai ô nhớ 31h và 32h. Ví dụ, nếu (30h) = A4h thì (31h) = 41h và (32h) = 34h.**

```

org 00h
        sjmp main
main:
        mov r0,#30h
        mov r2,#0
        mov a,@r0
        anl a,#0f0h
        swap a
        acall kiemtra
        mov a,@r0
        anl a,#0fh
        acall kiemtra
        sjmp exit

```

```

kiemtra:
        cjne a,#10,kiemtracocy
        sjmp cong37h
kiemtracocy:
        jc cong30h
        sjmp cong37h
cong30h:
        add a,#30h
        cjne r2,#1,luuvao31h
        mov 32h,a
        ret
luuvao31h:
        mov 31h,a
        inc r1
        ret
cong37h:
        add a,#37h
        cjne r1,#1,luuvao31h
        mov 32h,a
        ret
exit:

```



```

        nop
end

```

**20. Vẽ lưu đồ và viết chương trình con sosanh16 so sánh số 16 bit không dấu cất trong R7\_R6 (R7 chứa byte cao, R6 chứa byte thấp) với số 16 bit không dấu làm giá trị chuẩn cất trong ô nhớ 31h\_30h (31h chứa byte cao, 30h chứa byte thấp). kết quả trả về:**

- ( C ) = 1 nếu (R7\_R6)<(31h\_31H)  
 -( C )= 0 nếu (R7\_R6)>=(31h\_30h)

```

sosanh16:
    mov a,r7
    cjne a,31h,exit
bang:
    clr a
    mov a,r6
    cjne a,30h,exit
    ret
exit:
    ret

```

**21. Vẽ lưu đồ và viết chương trình kiểm tra một khối dữ liệu trong Ram ngoài có chiều dài 200 byte bắt đầu từ địa chỉ 2000h xem có bao nhiêu byte có giá trị là 0. kết quả được lưu vào ô nhớ có địa chỉ 1fffh trong Ram ngoài.**

```

org 00h
    sjmp main
main:
    mov dptr,#2000h
    mov r0,#200
    mov r1,#0
loop:
    clr a
    movc a,@a+dptr
    cjne a,#0,next
    inc r1
next:
    inc dptr
    djnz r0,loop
    mov dptr #1fffh

```

```

    mov a,r1
    movx @dptr,a
end

```

**22. Vẽ lưu đồ và viết chương trình cộng nội dung của ô nhớ Ram ngoài 4000h và 4001h. Nếu kết quả vượt quá FFh thì ghi giá trị 80h vào ô nhớ Ram ngoài 4002h, nếu không thì ghi giá trị 01h vào ô nhớ Ram ngoài 4002h.**

```

org 00h
    sjmp main

main:
    mov dptr,#4000h
    movc a,@a+dptr
    mov r0,a
    inc dptr
    clr a
    movc a,@a+dptr
    add a,r0
    jc ghi80h
    sjmp ghi01h
ghi80h:
    mov dptr,#4002h
    mov a,#80h
    movx @dptr,a
    sjmp exit
ghi01h:
    mov dptr,#4002h
    mov a,#01h
    movx @dptr,a
exit:
    nop
end

```

**23.Vẽ lưu đồ và viết chương trình cộng nội dung của hai ô nhớ RAM ngoài 2000H và 2001H. Byte cao của kết quả lưu vào ô nhớ 2003H, byte thấp của kết quả lưu vào ô nhớ 2002H.**

```

org 00h

```

```

mov dptr,#2000h
movc a,@a+dptr
mov r1,a
clr a
mov dptr,#2001h
movc a,@a+dptr
add a,r1
jc conho
mov dptr,#2002h
movx @dptr,a
sjmp exit
conho:
mov dptr,#2002h
movx @dptr,a
inc dptr
mov a,#1
movx @dptr,a
exit:
nop
end

```

**24. Vẽ lưu đồ và viết chương trình so sánh nội dung của hai ô nhớ RAM ngoài 1000H và 1001H. Cát giá trị lớn hơn trong hai số này vào ô nhớ 1002H.**

```

org 00h
mov dptr,#1000h
movc a,@a+dptr
mov b,a
inc dptr
clr a
mov a,@a+dptr
mov dptr,#1002h
cjne a,b,kiemtracocy
sjmp exit
kiemtracocy:
jc 1000hlonhon
movx @dptr,a
sjmp exit
1000hlonhon:
movx @dptr,a
exit:
nop

```

end

**25. Vẽ lưu đồ và viết chương trình nhân nội dung hai ô nhớ Ram ngoài 2000h và 2002h. byte cao của kết quả lưu vào ô nhớ 2003h, byte thấp của kết quả lưu vào ô nhớ 2002h**

```

org 00h
sjmp main
main:
mov dptr,#2000h
movc a,@a+dptr
mov b,a
clr a
inc dptr
movc a,@a+dptr
mul ab
mov dptr,#2002h
movx @dptr,a
mov a,b
inc dptr
movx @dptr,a
end

```

#### Câu 26:

**Vẽ lưu đồ và viết chương trình tăng dần giá trị nhị phân trong ô nhớ Ram ngoài có địa chỉ là 2000h sau mỗi khoảng thời gian 1s. Giả sử hệ thống dùng thạch anh 6MHz.**

```

org 00h
sjmp main
main:
mov dptr,#2000h
mov tmod,#20h
mov r7,#10
loop:
movc a,@a+dptr
acall delay1s
inc a
movx @dptr,a
sjmp loop

```

```

delay1s:
    mov th1,#high(-50000)
    mov tl1,#low(-50000)
    setb tr1
    djnz r7,delay1s
    mov r7,#10
    ret
end

```

### Câu 27:

Vẽ lưu đồ và viết chương trình tăng dần giá trị nhị phân trong ô nhớ Ram ngoài có địa chỉ là 1000h sau mỗi khoảng thời gian 2s. Giả sử hệ thống dùng thạch anh 6MHz.

```

org 00h
    sjmp main
main:
    mov dptr,#1000h
    mov tmod,#20h
    mov r7,#20
loop:
    movc a,@a+dptr
    acall delay2s
    inc a
    movx @dptr,a
    sjmp loop
delay2s:
    mov th1,#high(-50000)
    mov tl1,#low(-50000)
    setb tr1
    djnz r7,delay2s
    mov r7,#20
    ret
end

```

28.Vẽ lưu đồ và viết chương trình kiểm tra một khối dữ liệu trong RAM ngoài có chiều dài 100 byte bắt đầu từ địa chỉ 1000H xem có bao nhiêu byte có giá trị là 0, dương và âm. Kết quả được lưu vào các ô nhớ trong RAM nội như sau: ô

nhớ 20H chứa số byte bằng 0, ô nhớ 21H chứa số byte dương, ô nhớ 22H chứa số byte âm.

```

org 00h
    mov dptr,#1000h
    mov r3,#10
    sjmp main
main:
    clr a
    movc a,@a+dptr
kiemtra:
    cjne a,#0,khackhong
    inc r0
    sjmp lap
khackhong:
    rlc a
    jc soam
    sjmp soduong
soam:
    inc r1
    sjmp lap
soduong:
    inc r2
    sjmp lap
lap:
    inc dptr
    djnz r3,main
    mov 20h,r0
    mov 21h,r2
    mov 22h,r1
end

```

32.Vẽ lưu đồ và viết chương trình con tên BCD\_CHECK nhận dạng số BCD. Nếu nội dung thanh ghi A là số BCD nén (2 digit) thì chương trình con sẽ trả về cờ C = 1, ngược lại thì sẽ trả về cờ C = 0.

```

bcd_check:
    mov r1,a
    anl a,#0f0h
    swap a
    cjne a,#10,khac10

```

```
clr cy
ret
```

khac10:

```
jc nhohon10
clr cy
ret
```

nhohon10:

```
mov a,r1
anl a,#0fh
cjne a,#10,khac10lan2
clr cy
ret
```

khac10lan2:

```
jc ketluan
clr cy
ret
```

ketluan:

```
setb cy
ret
```

**29. Vẽ lưu đồ và viết chương trình xuất chuỗi xung vuông ra chân P1.0 với tần số  $f = 1\text{KHz}$ , duty cycle = 30% (thời gian mức cao = 30% thời gian của chu kỳ xung). Giả sử  $f_{\text{osc}} = 24\text{MHz}$ .**

```
ORG 0000H
LOOP:
SETB P1.0
ACALL DELAY600MC
CLR P1.0
ACALL DELAY1400MC
SJMP LOOP
DELAY600MC:
MOV R7,#2
LOOP1:
MOV R6,#150
DJNZ R6,$
DJNZ R7,LOOP1
RET
DELAY1400MC:
MOV R7,#7
LOOP2:
MOV R6,#100
```

```
DJNZ R6,$
DJNZ R7,LOOP2
RET
```

**33. Vẽ lưu đồ và viết một chương trình con tên store lưu trữ giá trị trong thanh ghi A vào Ram ngoài bắt đầu tại địa chỉ 1000h và theo thứ tự tăng dần địa chỉ (mỗi lần gọi store sẽ lưu một byte và địa chỉ bộ nhớ tăng thêm 1). cho dung lượng vùng nhớ là 100 byte lần lưu trữ thứ 101 sẽ được ghi đè vào địa chỉ đầu trở lại.**

loop:

```
mov r0,#0
mov dptr,#1000h
```

store:

```
movx @dptr,a
inc dptr
inc r0
cjne r0,#101,kiemtracocy
mov dptr,#1000h
sjmp store
```

kiemtracocy:

```
jc store
ret
```

Câu 37:

```
org 00h
sjmp main
```

main:

```
mov dptr,#1000h
mov r1,#10
```

loop:

```
clr a
movc a,@a+dptr
cjne a,#0,dem
sjmp boqua
```

dem:

```
inc r7
```

boqua:

```
inc dptr
djnz r1,loop
```

```

mov a,r7
mov dptr,#0fffh
movx @dptr,a
end

```

**40. Vẽ lưu đồ và viết chương trình con để tìm số lớn nhất ( không dấu) trong chuỗi dữ liệu cất trong vùng Ram nội từ địa chỉ 30h. chiều dài chuỗi là nội dung ô nhớ 2Fh. kết quả số lớn nhất cất vào ô nhớ 2Eh. Nếu giá trị chiều dài chuỗi bằng 0 thì kết quả là 00h.**

```

timmax:
    mov r0,#30h
    mov a,@2fh
    dec a
    mov r1,a
    cjne a,#0,ketthuc

```

```

main:
    mov a,@r0
loop:
    inc r0
    mov b,@r0
    cjne a,b,kiemtra
    sjmp tieptuc

```

```

kiemtra:
    jnc tieptuc
    mov a,b

```

```

tieptuc:
    djnz r1,loop
    mov 2eh,a
    ret

```

```

ketthuc:
    mov 2eh,#0
    ret

```