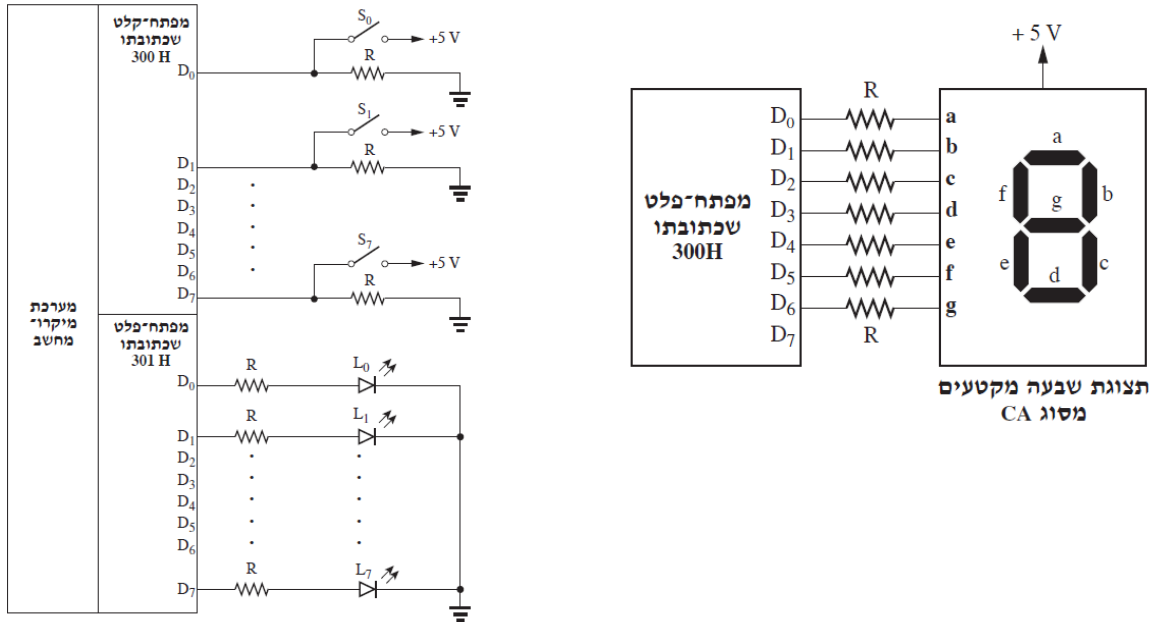


שאלות ותשובות התקני קלט/פלט שפת C ואסמבלר

כל השאלות מתייחסות לשרטוטי קלט פלט הבאים:



1. כתוב תוכנית אשר תקלוט את מצב מפסק S0 מכתובת 300H, אם הוא פתוח תדליק את L0 ותכבה את L7 בכתובת 301H, אם הוא סגור תכבה את L0 ותדליק את L7. שאר הלדים כבויים תמיד.
2. כתוב תוכנית שתקלוט את מצב המפסקים. אם הנתון המיוצג על ידי ארבעת המפסקים הנמוכים (S0-S3) גבוה או שווה מהנתון המיוצג על ידי ארבעת המפסקים הגבוהים (S4-S7), הדלק את ארבעת הלדים L0-L3 אחרת הדלק את הלדים L3-L7.
3. כתוב תוכנית אשר תקלוט את מצב מפסק S0 מכתובת 300H, אם הוא פתוח תציג על תצוגת ה- seven segment את הספרה 0 אחרת תציג 1.
4. כתוב תוכנית אשר תגרום ללדים להידלק אחד אחרי השני 16 פעם. השהייה בין הדלקה להדלקה שנייה אחת.
5. הוסף קטע לתוכנית 4 אשר יבצע את התוכנית רק אם מפסק S7 פתוח, אחרת התוכנית תמתין לפתיחתו.
6. כתוב תוכנית אשר תדליק אחד אחרי השני ספרות בתצוגה מ-0 עד 9. באופן מחזורי כל עוד מפסק S1 פתוח. השהייה בין הדלקה להדלקה שנייה אחת.  
הערה: יש להשתמש במערך.
7. כתוב תוכנית אשר תמנה כמה מפסקים פתוחים, אם המספר גדול שווה מ-4 הדלק את כל הלדים אחרת כבה אותם.
8. כתוב תוכנית אשר תמנה כמה מפסקים סגורים ותדליק לדים באותה כמות. למשל אם שלושה מפסקים סגורים L0, L1, L2 יהיו דלוקים.

## תשובות

1. \_\_מפסק פתוח '0'.

```
#include<windows.h>
#include<stdio.h>

void main()
{
    char in;
    in=Inp32(0x300);
    in=in&0x01;
    if(in==0)
        Out32(0x301,0x01);
    else
        Out32(0x301,0x80);
}
```

### ASM

```
MOV DX,300H
IN AL,DX
MOV DX,301H
AND AL,00000001B
JZ OPEN
MOV AL,01H
JMP LED
OPEN: MOV AL,80H
LED:  OUT DX,AL
RET
```

2.

```
#include<windows.h>
#include<stdio.h>

void main()
{
    char in_l,in_h;
    in_l=Inp32(0x300);
    in_h=in_l;
    in_l=in_l&0x0f;
    in_h=in_h>>4;
    if(in_l>=in_h)
        Out32(0x301,0x0f);
    else
        Out32(0x301,0xf0);
}
```

### ASM

```
MOV DX,300H
IN AL,DX
MOV AH,AL
MOV DX,301H
AND AL,00001111B
```

```

        AND AH,11110000H
        MOV CL,4
        ROR AH,CL
        CMP AL,AH
        JNC BIG
        MOV AL,0F0H
        JMP LED
BIG:    MOV AL,0FH
LED:    OUT DX,AL
        RET

```

3. שימו לב '0' מדליק את המקטע הרצוי בתצגה.

```

void main()
{
    char in;
    in=Inp32(0x300);
    in=in&0x01;
    if(in==0)
        Out32(0x301,0x40);
    else
        Out32(0x301,0x79);
}

```

ASM

```

        MOV DX,300H
        IN AL,DX
        MOV DX,301H
        AND AL,00000001B
        JZ OPEN
        MOV AL,79H
        JMP LED
OPEN:   MOV AL,40H
LED:    OUT DX,AL
        RET

```

4. באסמבלר השתמשתי בתוכנית משנה בשם DEL\_1S שהיא נתונה.

```

void main()
{
    int i,j;
    char L;
    for(j=0;j<2;j++)
    {
        for(i=0,L=0x01;i<8;i++)
        {
            Out32(0x301,L);
            Sleep(1000);
            L=L<<1;
        }
    }
}

```

ASM

```

        MOV CX,16
        MOV DX,301H
        MOV AL,1
AGAIN:   OUT DX,AL

```

```

CALL DEL_1S
ROL AL,1
LOOP AGAIN
RET

```

5. יש להוסיף את קטע התוכנית הבא כהוראות ראשונות בתוכנית

```

do{
    L=In32(0x300);
    L=L&0x80;
}
while(L!=0);

```

ASM

```

ST: MOV DX,300H
    IN AL,DX
    AND AL,10000000B
    JNZ ST

```

6. לא פתרתי באסמבלר משום שלא למדנו כיצד להגדיר מערכים.

```

void main()
{
    int led[10]={0x40,0x79,0x24,0x60,0x19,06,0x02,0x78,0x00,0x10};
    int i;
    char L;
    while(1)
    {
        do{
            L=Inp32(0x300);
            L=L&0x02;
        }
        while(L!=0);

        for(i=0,L=0x01;i<10;i++)
        {
            Out32(0x300,led[i]);
            Sleep(1000);
        }
    }
}

```

7.

```

void main()
{
    int i,c=0;
    char P;
    P=Inp32(0x301);
    for(i=0;i<8;i++)
    {
        if((P&0x01)==0)
            c++;
        P=P>>1;
    }
    if(c>=4)
        Out32(0x300,0xff);
    else
        Out32(0x300,00);
}

```

### ASM

```
MOV DX,301H
IN AL,DX
MOV AH,0
MOV CX,8
SS:   RCL AL,1
      JC DALEG
      INC AH
DALEG:LOOP SS
      CMP AH,4
      JNC BIG
      MOV AL,0
      JMP LL
BIG:  MOV AL,0xFF
LL:   MOV DX,300H
      OUT DX,AL
      RET
```

.8

```
void main()
{
    int i,c=0;
    char P,L=0x00;
    P=Inp32(0x301);
    for(i=0;i<8;i++)
    {
        if((P&0x01)==1)
            c++;
        P=P>>1;
    }
    for(L=0x01,i=1;i<c;i++)
        L=L<<1+1;

    Out32(0x300,L);
}
```

### ASM

```
MOV DX,300H
IN AL,DX
MOV AH,0
MOV CX,8
SS:   RCL AL,1
      JNC DALEG
      INC AH
DALEG:LOOP SS
      MOV AL,0
      CMP AH,0
      JZ LL
TT:   STC
      RCL AL,1
      DEC AH
      JNZ TT
      MOV AL,AH

LL:   MOV DX,300H
      OUT DX,AL
      RET
```

