

## הוראות וסימנים מוסכמים - C51

### הכרזה על משתנים.

char - משתנה המכיל 8 סיביות, סיבית אחרונה סיבית סימן. -+ 127.  
int - משתנה המכיל 16 סיביות סיבית אחרונה סיבית סימן + 32628.  
unsigned char - גורם למשתנה להכיל מספר חיובי בלבד. + 255.  
unsigned int - גורם למשתנה להכיל רק מספר חיובי 65356 +.  
[4] A (int, char) - הכרזה על מערך בשם A בעל 4 תאים.  
[4] A (int, char) - אתחול נתונים עם ההכרזה על המערך.

0x2000 #define mark - הכרזה על קבוע, שימוש ב-mark הוא שווה ערך לשימוש בנתון 0x2000

### סימנים של פעולות אריתמטיות -

סדר הפעולות הוא כמו בחוקי החשבון.

\* כפל  
/ חילוק  
% מודולו (שארית)  
+ חיבור  
- חיסור

### הוראות.

#### הוראות תנאי if

if (תנאי לוגי) אם מתקיים התנאי סדרת ההוראות המוגדרות בסוגריים מתקיימות, אם לא, התוכנית מדלגת על הסוגריים.

{  
הסימנים לשאלות תנאי: < גדול  
> קטן  
=< גדול שווה  
=> קטן שווה  
== שווה  
!= שונה

משפטים לוגיים :

&& וגם  
|| או

#### הוראות תנאי if-else

אם מתקיים התנאי הלוגי תבצע ההוראות בסוגריים, אם לא, תבצע ההוראות בסוגריים תחת ה- else.

(.תנאי לוגי) if  
{  
סדרת הוראות....  
}  
else  
{  
סדרת הוראות....  
}

## הוראת switch

הוראת תנאי זאת שימושית כאשר ההשוואה היא למספר בודד. במידה והמשתנה  $1 = x$  התוכנית תבצע את ההוראות הרשומות ב-case 1 עד להוראת break תוציא את התוכנית מה-switch. אם  $x = 2$  התוכנית תכנס ל-case 2 וכן הלאה.

```
switch (x)
{
case 1:

break;
case 2:

break;
case 3:

break;
}
```

## הוראות לולאה while

(תנאי לוגי) while אם מתקיים התנאי התוכנית מבצעת סדרת הוראות שבסוגריים ובסיומה היא שוב בודקת את התנאי, וכן הלאה. כאשר התנאי לא מתקיים התוכנית תצא מהלולאה ותמשיך בתוכנית.

```
.....סדרת הוראות.
{
```

## הוראות לולאה do-while

לולאה זאת מתבצעת לפני בדיקת התנאי, ורק בסיומה מתבצעת בדיקת התנאי. אם הוא מתקיים התוכנית חוזרת ללולאה ואם לא התוכנית ממשיכה.

```
do
{
.....סדרת הוראות.
}
```

while (תנאי לוגי);

## הוראות לולאה for

(עידכון מונה; תנאי יציאה; תנאי התחלה) for למשל:

לולאת for יותר אלגנטית.  $i = 0$  הוא תנאי ההתחלה של הלולאה, ורק בפעם הראשונה  $i$  מקבל את הערך 0.  $i < 10$  תנאי יציאה מהלולאה כאשר התנאי יתקיים התוכנית תבצע את הלולאה, במידה והתנאי לא מתקיים התוכנית תצא מהלולאה ותמשיך בתוכנית.  $i++$  הגדלת המונה באחד בכל ביצוע לולאה.

```
for(i=0;i<10;i++)
{
.....סדרת הוראות.
}
```

## המשפט continue

משפט המופיע בלולאות בד"כ, בזמן ביצוע המשפט מתבצעת חזרה על הלולאה ממקום משפט זה. (תנאי לוגי) while הלולאה תתבצע עד משפט ה-continue ותחזור לבדוק את תנאי הלולאה.

```
{
continue;
}
```

## פעולות לוגיות

^ xor  
& and  
| or  
~ not  
>> rr דוגמא: a=a<<3 הזז את a 3 פעמים שמאלה.  
<< rl

ניתן לבצע פעולות לוגיות בין סיביות.

## פניה לזיכרון – הוראות ייחודיות ל-C51

XBYTE[addr]=a פנייה לזיכרון חיצוני, העתק את הערך שב-a לזיכרון או להתקן שבכתובת .addr  
DBYTE [addr]=a פנייה לזיכרון הפנימי.

דוגמאות להכרזה על משתנים בכתובות קבועות

```
1000h בתא t1 מוגדר בתא ;xdata int t1 _at_ 0x1000
משתנה t1 מוגדר בתא 1000h
2000h בתא t2[10] מוגדר בתא ;xdata int t2[10] _at_ 0x2000
הכרזה על מערך t2 של 10 תאים החל מכתובת 2000h
0x20 בתא t3[10] מוגדר בתא ;data char t3[10] _at_ 0x20
הכרזה על מערך בזיכרון הפנימי
0x100 בתא d מוגדר בתא ;at 0x100 xdata int d
צורה נוספת של הכרזה על משתנה בזיכרון החיצוני
```

פנייה לאוגרים הייעודיים הוא בשם המקורי כמו באסמבלר.  
למשל פנייה ל-TMOD.  
TMOD=0X52;

### פנייה ל-PORT

P1=0X0F; - שליחה של 8 סיביות לכל ה-PORT ניתן לפנות כך ל-P0,P1,P2,P3.

### פנייה לסיבית אחת ב-PORT

יש להכריז על הסיבית הראשית התוכנית באופן הבא:  
sbit name= P1^0;  
הכרזה על סיבית ב-P1.0, בשם name כעת ניתן לפנות לסיבית באמצעות פנייה למשתנה name.

### פונקציות

1. פונקציה פשוטה -. function\_name() יגרום לביצוע ההוראות.

```
void function_name()
{
.....
סדרת הוראות
}
```

2. פונקציה המקבלת ערך – `function_name(a,b)` ישלח את `a` ל-`x`, ואת `b` ל-`y`.

```
Void function_name ( int x, char y ...)  
{  
    .....  
    סדרת הוראות  
    תוך שימוש במשתנים x,y  
}
```

3. פונקציה המחזירה ערך- `a=function_name()` ביצוע הפונקציה וערך `f` יעבור ל-`a`.

```
int function_name ()  
{  
    int f;  
    .....  
    סדרת הוראות  
    return f;  
}
```