

הוראות וסימנים מוסכמים C-

הכרזה על משתנים.

טבלת הטיפוסים

טיפוס	מספר בתים	תחום המספרים
char, signed char	1	128 - 127+
unsigned char	1	0 - 255+
int, sign int	2	32767+ 32768-
unsigned int	2	65535+ 0
long, long int, signed long int	4	2147483647+ 2147483648-
unsigned long, unsigned long int	4	4294967295+ 0
float (7 ספרות דיוק)	4	3.4e+38 3.4e-38
double (15 ספרות דיוק)	8	1.7e+308 1.7e-308
long double (19 ספרות דיוק)	10	3.4e+4932 3.4e-4932

פעולות אריתמטיות –

אופרטורים חשבוניים - מאפשרים לבצע פעולות חשבוניות כמו חיבור, חיסור, חילוק, ומודולו (שארית):

$a+b$ חיבור המשתנים a ו-b.

$a-b$ חיסור המשתנה a ב-b.

$a*b$ מכפלת המשתנים a ו-b.

$a\%b$ שארית של המשתנה a בחלוקה ב-b.

פעולות לוגיות

OR	
AND	&
NOT	~
XOR	^
ROR	>>
ROL	<<

משפטי תנאי

אופרטורי יחס

$<$ יחס קטן (מכיוון שסדר ביצוע הפעולות הוא משמאל לימין)

$>$ יחס גדול.

$==$ יחס שיוויון.

$<=$ יחס גדול או שווה.

$=>$ יחס קטן או שווה.

אופרטורים לוגיים (ערך 1 מציין אמת 0 מציין שקר)

$\&\&$ פירושו וגם.

$\|\|$ פירושו או.

$!=$ פירושו שונה.

! פירושו לא.

סדר הקדימויות מהגבוה לנמוך

++, --, !	1
*, %, /	2
+, -	3
<, <=, >, >=	4
==, !=	5
&&	6
	7
=, +=, -=, /=, *	8

הוראות.

פקודות לביצוע מותנה

הפקודות if - else

הפקודות switch - case

האופרטור ?

פקודות לביצוע לולאות

הפקודות for

הפקודה while

הפקודה do while

(ופקודות עזר break - continue)

פקודת if

הרויח של הפקודה if

```
דוגמא:
if (a>b)
{
    printf(" a gadol" );
}
else
{
    printf (" b gadol ");
}
```

```
if (term)
{
    expression a;
}
else
{
    expression b;
}
```

הערות:

לא חובה לצרף else להוראה.

בהוראה אחת אחרי התנאי לא חובה לפתוח סוגריים מסולסלות.

התנאי הלוגי יכול להיות מורכב מכמה תנאים על פי הכללים.

תבנית לשימוש בהוראה else if

```
דוגמא:  
if (a<55)  
{  
    printf(" FAIL ");  
}  
else if ( a >=55 && a<80 )  
{  
    printf (" GOOD ");  
}  
else  
    Printf ( " VERY GOOD");
```

```
If (exp)  
{  
    ...  
}  
else if(exp)  
{  
    ...  
}  
else if(exp)  
{  
    ...  
}  
else  
{  
    ...  
}
```

הערות
יתבצע רק תנאי אחד, הראשון שיתקיים.
סדר הבדיקה על פי סדר הכתיבה מלמעלה למטה.

הוראת switch

```
דוגמא:  
switch (x)  
{  
    case 1 : printf(" x=1 ");  
             break;  
    case 2: printf(" x=2");  
             break;  
    case 6: printf (" x=6");  
             break;  
    default : printf (" not found");  
              break;  
}
```

```
switch(exp or variable)  
{  
    case constant1: command; break;  
    case constant2: command; break;  
    case constant3: command;break;  
  
    case constantn: command;break;  
    default: command;  
}
```

הערות:
constant קבוע מספרי
break מוציא את התוכנית מה- switch
default לא חובה.
אם $x = 2$ התוכנית מדפיסה $x=2$ ויצאת מה- switch .

הוראות לולאה while

```
דוגמא:  
1 #include <stdio.h>  
2  
3 void main()  
4 {  
5     int times = 10;  
6     printf("Printing the numbers form  
7 10 to 1: \n\r");  
8     while (times > 0)  
9     {  
10        printf ("%d ", times);  
11        times--;  
12    }  
13 }
```

```
while(boolean condition)  
{  
    .....  
}
```

הערות:
אם התנאי הלוגי בסוגריים מתקיים הלולאה מתבצעת פעם אחת ושוב בודקת את התנאי וכן הלאה, כאשר הלולאה לא מתבצעת ממשיכה התוכנית.

הוראות לולאה do- while

```
1 #include <stdio.h>
2
3 void main()
4 {
5     int result = 0;
6     printf("The sq of the number from 1 to 10: \n\r");
7     do
8     {
9         result += 1;
10        printf("%d \n\r", result*result);
11    }
12    while (result < 10);
13 }
```

דוגמא:

```
do
{
...
}
while (condition)
```

ההוראה מבצעת את מה שבסוגריים ולאחר מכן מבצעת בדיקת הלולאה. אם התנאי מתקיים ההוראה חוזרת על הלולאה אם לא ממשיכים.

הוראות לולאה for

הלולאה בנויה משלושה חלקים : אתחול משתנים, תנאי ביצוע, קידום משתנים. בחלק הראשון מתבצע איתחול משתנים. בחלק השני מתבצעת בדיקת התנאי אם הוא אמת הלולאה מתבצעת, לאחר ביצוע הלולאה מתבצע קידום משתנים. אפשר לוותר על כל חלק בהוראה.

```
1 #include <stdio.h>
2 void main()
3 {
4     int times;
5     printf("Printing the numbers form 10 to 1: \n\r");
6     for (times = 10 ; times > 0 ; times--)
7         printf ("%d ", times);
8 }
```

דוגמא:

```
for (initialization ; boolean condition ; increment)
{
...
}
```

פקודות break , continue

פקודות אלה קשורות להוראת הלולאה. בעזרת ההוראה break ניתן לצאת מלולאה, הפקודה continue מדלגת על הוראת עד ביצוע הבא של הלולאה.

```
if (condition)
break;
.....
```

אם מתקיים התנאי יציאה מהלולאה.

```
if (condition)
continue;
.....
```

אם מתקיים התנאי דלג על המשך ההוראות וחזור לתחילת הלולאה.

מערכים

מערך הוא רשימה של משתנים המסודרים בסדר עולה ופונים אליהם באמצעות שם ואינדקס.
char array[10] - הכרזה על מערך של עשרה תאים בשם array כל תא char .
char[0]=22 – הכנסת המספר 22 למיקום הראשון במערך.
int data[4]={ 12,34,56,00} - הכרזה על מערך וטעינת נתונים למערך.

מבנה

מה שמאפיין מבנה שבשם אחד ניתן להכיל משתנים מטיפוסים שונים.

<pre>struct address { char city[20]; char street[20]; int num_house; long code_area; };</pre>	דוגמא	<pre>struct name_of_struct { variables };</pre>
---	-------	---

אתחול משתנים למבנה:

gets (address.city) – קליטת מחרוזת למערך שבתוך המיבנה.
address.num_house=1916 – טעינת מספר למשתנה שבמערך.

מערכים של מבנים:

struct address st[5] – יצרנו חמש מבנים מסוג address
st[1].num_house=2067 כתיבה למשתנה num_house במבנה השני .
st[4].code_area=12345678 – כתיבה למשתנה code_area במבנה החמישי.

מצביעים

מצביע הוא משתנה המכיל נתון מסוג כתובת.
גודל המשתנה קבוע בהכרזה על מצביע מודיעים על איזה טיפוס של משתנה הוא מצביע

הגדרת משתנה מסוג מצביע

int *px - הכרזה על משתנה מסוג מצביע, אשר יצביע על משתנה מסוג int
char *pw – הכרזה על משתנה מסוג מצביע, אשר יצביע על משתנה מסוג char

אתחול מצביע והשמה

int x, *px - הכרזה על משתנה x ומצביע px .
xp=&x - המצביע מקבל את הכתובת של המשתנה x .
*xp=6 - טעינת 6 של משתנה x . שווה ערך להוראה x=6 .

הערה: לכוכבית * שני תפקידים. בהכרזה הכוכבית מודיעה על משתנה מסוג מצביע.
בפעולות כוכבית לפני המצביע גורמת לפעולה להתבצע על המשתנה.

מצביעים ומערכים

התא הראשון במערך נמצא בכתובת הראשונה של המערך, לפיכך ניתן להעביר את הכתובת למצביע
ובכך לעבוד עם המערך.

דוגמא:

```
int array[10] - הכרזה על מערך.  
int *parray - הכרזה על מצביע.  
parray=&array[0] – טעינת הכתובת הראשונה של המערך למצביע. שווה ערך parray=array  
הערה: שם המערך הוא הכתובת הראשונה שלו.
```

פונקציות בנויות של השפה

פונקציות קלט\פלט.

- putchar('a') - פלט של תו בודד.
- c=getchar() - קלט של תו בודד למשתנה c.
- printf(".....") - הדפסה של מחרוזת.
- printf("%d %c%f%s",a,b,c,d) - הדפסה של משתנים מסוגים שונים לפי תווי הבקרה הבאים:
 - d - מספר דצימלי
 - c - תו
 - f - מספר מסוג float מספרים לא שלמים.
 - s - מחרוזת
- scanf("%d",&x) - קליטת נתון מהמקלדת למשתנה x בהצגה דצימלית.

פונקציות טיפול במחרוזות

- gets(שם_מערך) - קליטת מחרוזת לתוך מערך מוגדר.
- puts(שם_מערך) - הדפסת מחרוזת מתוך מערך.
- x= strlen(שם_מערך) - מציאת אורך המחרוזת והצבת התוצאה ב-x.
- strcpy(מחרוזת_מקור, מחרוזת_יעד) - מחרוזת המקור מועתקת למחרוזת היעד.
- strcat(מחרוזת_מקור, מחרוזת_יעד) - מחרוזת המקור מתווספת לסוף מחרוזת היעד.

פונקציות קבצי טקסט

לפני עבודה עם קבצי טקסט יש להגדיר משתנה מסוג מצביע למבנה ה-FILE

- FILE *fp - משתנה מסוג מצביע למבנה FILE.
- fp = fopen(שם הקובץ) - פתיחה של קובץ, סוג בגישה קריאה \ כתיבה וכו', המצביע fp מקבל NULL אם יש בעיה בפתיחה אחרת ניתן לעבוד עם הקובץ.
- fclose(FILE_מבנה) - סגירת הקובץ לאחר עבודה איתו.
- c = fgetc(FILE_מבנה) - קריאת תו מקובץ טקסט.
- fputc(תו, FILE_מבנה) - כתיבת תו לקובץ טקסט.
- fgets(FILE_מבנה, אורך_מירבי, מחרוזת) - קריאת שורה מקובץ למחרוזת, אורך מרבי של המחרוזת מוגדר בפונקציה. סיום שורה מזהה על ידי 'n'.
- fputs(מחרוזת, FILE_מבנה) - כותבים את המחרוזת לפונקציה.
- fprintf(FILE_מבנה, מצביע_למבנה, מחרוזת בקרה, מצביע למבנה FILE מודפסת לקובץ שהמצביע למבנה FILE מצביע עליה.

סוגי גישה

- r - קריאה בלבד
- w - כתיבה בלבד
- a - הוספה לקובץ קיים

פניה להתקני קלט פלט

- outportb(address, data) - כתובת ההתקן, data - מידע הנשלח להתקן
- x=inportb(address) - קלוט נתון מהתקן קלט שכתובתו address למשתנה x.

פונקציות

1. פונקציה שלא מקבלת ערך ולא מחזירה

```
void function ()
{
    .....
    .....
}
```

2. פונקציה שמקבלת ערכים.

```
void function(int x, char y, ....)
{
    .....
    .....
}.
```

יש להגדיר את המשתנים אליהם נשלח הערך בתוך ההכרזה על הפונקציה. בדוגמא הפו' תקבל שני ערכים ל- x ו- y .

3. פונקציה המחזירה ערך.

ניתן לשלב פונקציה המקבלת ערך ומחזירה.

```
int function ()
{
    int x;
    .....
    .....
    return x ;
}
```

פונקציות מובנות קלט\פלט

`printf("hello %d ",a)` – הדפסה על המסך מחרוזת שבגרשיים, אם אפשרות להדפסה של ערך משתנה. בדוגמא: הערך של המשתנה `a` יופיע כערך דצימלי על המסך. סימנים נוספים: `%c` – הצגת תו, `%f` – הצגת מספר לא שלם, `%s` – הדפסת מחרוזת.
`scanf("%d",&a)` – קליטת מספר מהמקלדת עם הד למסך. הערך יכנס למשתנה `a`.
`a=getch()` – קליטת ערך ASCII של תו בודד למשתנה `a`.
`putch(a)` – הדפסה של תו בודד אשר ערך ה-ASCII שלו נמצא במשתנה `a`.
`clrscr()` – ניקוי המסך.
`gotoxy(x,y)` – קביעת הסמן על המסך.
`delay(1000)` – ביצוע השהיה הערך בסוגרים ב-`ms` (מילי שניות).