

גירסה 1.00 – 29.02.2004

# לולאות בשפת C

ניר אדר

מסמך זה הורד מהאתר <http://underwar.livedns.co.il>

אין להפיץ מסמך זה במדיה כלשהי, ללא אישור מפורש מאת המחבר.

מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר עשה את מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר.

כל הזכויות שמורות לניר אדר

Nir Adar

Email: [underwar@hotmail.com](mailto:underwar@hotmail.com)

Home Page: <http://underwar.livedns.co.il>

### לולאות בשפת C

במסמך נדגים שימוש בלולאות בשפת C. כמו כן נשתמש בפקודות שונות המיוחדות ל-Borland C, על מנת לעצב את פלט התוכנית שלנו.

מסמך זה מיועד למתחילים הבאים להכיר את שפת C, והוא השיעור השני בסדרת המסמכים הבסיסיים אותה אני כותב, המלמדים את יסודות C עם דוגמאות ספציפיות ל-Borland C.

נציג כעת את הפקודות בהן נשתמש בדוגמאות במסמך זה. הפקודות ממוינות לפי ה-include שצריך לעשות כדי להשתמש בהן:

#### stdlib.h

דוגמא לשימוש	פעולה	הפקודה
randomize();	מאתחלת את מנוע המספרים האקראיים של C. אם אנחנו משתמשים בפונקציה random בתוכנית שלנו, עלינו לקרוא פעם אחת לפונקציה זו.	randomize()
/* x will have random value from 1 to 10 */ x = random(10) + 1;	מחזירה מספר שלם חיובי בין 0 ל-X.	random(X)

#### dos.h

דוגמא לשימוש	פעולה	הפקודה
/* delay for one second */ delay(1000);	עוצרת את התוכנית ל-x אלפיות שניה.	delay(x)

**conio.h**

דוגמא לשימוש	פעולה	הפקודה
clrscr();	מוחקת את כל הטקסט מהמסך	clrscr()
getch();	עוצרת את התוכנית ומחכה שהמשתמש ילחץ על מקש כלשהו. עם לחיצת המקש התוכנית ממשיכה.	getch()
gotoxy(10, 20);	מעבירה את סמן הטקסט לנקודה x, y על המסך. הטקסט הבא שיודפס, על ידי printf למשל, יופיע החל מנקודה זו. במצב טקסט גודל המסך הוא 80 × 24.	gotoxy(x, y)
textcolor(4); textcolor(BLUE);	משנה את צבע הטקסט. יש להשתמש ב-cprintf כדי שהטקסט אכן יודפס בצבע המבוקש. טבלת הצבעים – מיד. ניתן להשתמש במספר של הצבע או בצבע עצמו כמו שהוא מופיע בטבלה.	textcolor(color)
cprintf("Hello, World\n");	דומה ל-printf, אבל במידה והחלנו עיצובים על הטקסט, כגון צבע, פקודה זו תציג את הטקסט מעוצב.	cprintf(string, ...)

טבלת הצבעים עבור textcolor:

צבע	מספר המייצג את הצבע
BLACK	0
BLUE	1
GREEN	2
CYAN	3
RED	4
MAGENTA	5
BROWN	6
LIGHTGRAY	7
DARKGRAY	8
LIGHTBLUE	9
LIGHTGREEN	10
LIGHTCYAN	11
LIGHTRED	12
LIGHTMAGENTA	13
YELLOW	14
WHITE	15

## לולאת while

התחביר של לולאת while:

```
while (condition)
{
}

```

עם הכניסה ללולאה, נבדק הביטוי condition. אם ערכו אמת, נבצע את הקטע שבין הסוגריים המסולסלות. בסיום הקטע נבדוק שוב את התנאי condition, ואם הוא ממשיך להתקיים נחזור על התהליך - נבצע את הפקודות שבסוגריים המסולסלות ונחזור שוב לבדוק את condition. משמעות לולאה זו היא בעצם "כל עוד condition מתקיים, בצע את הביטוי שבין הסוגריים המסולסלות".

## תוכנית ראשונה

המטרה: לקבל מספר X מהמשתמש ולהדפיס את המספרים מ-1 עד אותו מספר. השלבים:

1. נקלוט מספר X מן המשתמש.
2. נגדיר מונה ונשים בו 1.
3. כל עוד המונה קטן מ-X:
  - a. נדפיס את המונה.
  - b. נגדיל את המונה ב-1.

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int mone = 1;
    int x;

    printf("Please enter a number: ");
    scanf("%d", &x);

    while (mone <= x)
    {
        printf("%d ", mone);
        mone = mone + 1;
    }
    printf("\n");
    getch();
    return 0;
}

```

**תוכנית 2**

לולאה שמדפיסה 5 מספרים אקראיים בין 1 אל המספר אותו המשתמש בחר. נשים לב שמבנה הלולאה איננו משתנה. אנחנו מגדירים מונה ומגדילים את הערך שלנו מ-1 עד 5. ההבדל הוא שבמקום להדפיס את המונה הפעם אנחנו מדפיסים משהו אחר – מספר אקראי שהוא בין 1 ל-X.

1. נקלוט מספר X מן המשתמש.

2. נגדיר מונה ונשים בו 1.

3. כל עוד המונה קטן מ-X:

a. הדפס מספר אקראי בין 1 ל-X.

b. נגדיל את המונה ב-1.

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int i = 1;
    int x;

    printf("Please enter a number: ");
    scanf("%d", &x);
    randomize();

    while (i <= 5)
    {
        printf("%d ", random(x)+1);
        i = i+1;
    }
    printf("\n");
    getch();
    return 0;
}
```

**לולאת for**

התחביר של לולאת for:

```
for ([<initialize>; [<condition>]; [<step>])
{
    /* do something */
}
```

initialize, condition ו-step הם שלושה ביטויים. כאשר אנו משתמשים בפקודת for, ראשית מתבצע הביטוי initialize. לאחר מכן נבדק ערכו של הביטוי condition. אם ערכו של ביטוי זה הוא אמת, נבצע את הפקודות שבין הסוגריים המסולסלות. לאחר מכן, נבצע את הביטוי step, ונחזור לבדוק שוב את condition. בצורה כזו נמשיך: ביצוע הקוד שבין הסוגריים המסולסלות, ביצוע הביטוי step ובדיקת ערכו של condition. אם לא קיימים סוגריים מסולסלות אחרי ה-for, תבוצע הפקודה שאחרי ה-for.

### תוכנית 3

לולאות המדפיסות את 10 פעמים את המשפט "Hello, World". נציג שתי לולאות שיבצעו את אותו אלגוריתם, ונביט בהבדלים ביניהן מבחינת הכתיבה:

1. אתחל את x להיות 1.

2. כל עוד x קטן מ-10:

a. הדפס "Hello, World".

b. הגדל את x ב-1.

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

int main()
{
    int x;
    clrscr();

    /* On the first time, we use while loop */
    x = 1;
    while (x <= 10)
    {
        printf("Hello, World!\n");
        x = x + 1;
    }

    /* printing again, now using for */

    /* for (start; KOL OD (condition); step) */
    for (x = 1; x <= 10; x = x + 1)
    {
        printf("Hello, World!\n");
    }

    printf("\n");

    getch();
    return 0;
}
```

## תוכנית 4

נכתוב כעת תוכנית המדפיסה את המחרוזת "Hello, World" ב-15 צבעים שונים. אנחנו בוחרים צבעים על ידי מספר בין 0 ל-15, כאשר 0 זה צבע הרקע של המסך (שחור) ו-1 עד 15 הם הצבעים השונים האפשריים.

המשתנה x בתוכנית משמש גם כמונה ללולאה, וגם הוא הצבע שאנחנו בוחרים כל פעם עבור הטקסט שאנחנו מציגים.

שלבי הפעולה:

1. שים ב-x את הערך 1.
2. כל עוד x קטן או שווה מ-15:
  - a. שנה את צבע הטקסט לצבע ש-x מייצג.
  - b. הדפס "Hello, World".
  - c. הגדל את x ב-1.

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

int main()
{
    int x;
    clrscr();

    for (x = 1; x <= 15; x = x + 1)
    {
        textcolor(x);
        cprintf("Hello, World!\n");
    }

    printf("\n");

    getch();
    return 0;
}
```

## תוכנית 5

התוכנית הבאה באה להדגים את הפקודה gotoxy. פקודה זו מאפשרת לנו במצב של מסך טקסט לקפוץ לנקודה מסויימת במסך. גודל המסך במצב טקסט הוא בדרך כלל 80x24. כפרמטרים לפקודה אנחנו מעבירים את המיקום החדש אליו אנחנו רוצים לקפוץ.

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

int main()
{
    int x;
    clrscr();

    /* the screen size is 80x24 */
    gotoxy(30, 20);
    cprintf("Hi :)");

    getch();
    return 0;
}
```

## תוכנית 6

מטרת התוכנית: הדפסת טקסט 100 פעמים במקומות אקראיים על המסך ובצבעים אקראיים. הרעיון: ניצור לולאה שתרוץ 100 פעמים. בכל מחזור של הלולאה, נקפוץ למקום אקראי, נבחר צבע אקראי ונדפיס את הטקסט שאנחנו רוצים להדפיס.

שלבי הפעולה:

1. אתחל משתנה x בערך 1.
2. כל עוד x קטן או שווה ל-100:
  - a. קפוץ למקום אקראי על המסך.
  - b. בחר צבע אקראי.
  - c. הדפס את הטקסט המבוקש.
  - d. חכה עשירית שניה.
  - e. הגדל את x.

להלן הקוד המממש את התוכנית:

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>

int main()
{
    int x;
    randomize();
    clrscr();
    for (x = 1; x <= 100; x = x + 1)
    {
        gotoxy (random(80)+1, random(24)+1);
        textcolor (random(15)+1);
        cprintf("Hello, World");
        delay(100);
    }

    getch();
    return 0;
}
```

## תוכנית 7

מטרת התוכנית: טקסט "רוקד". נציג את המחרוזת "Hello, World" על המסך, וניתן לה לזוז כל פעם ממקום למקום על המסך. שלבי הפעולה:

1. בחר מקום התחלה אקראי.
2. בצע 100 פעמים:
  - a. בחר צבע אקראי.
  - b. קפוץ למקום שבחרנו והדפס שם את הטקסט.
  - c. חכה עשירית שניה, ואז חזור למקום ההתחלה ומחק את הטקסט (על ידי כתיבה שלו שוב בצבע שחור).
  - d. הוסף לקורדינטת x של ההדפסה מספר אקראי בין 1-ל-1. (בצורה זו או שנוזו צעד אחד שמאלה, או צעד אחד ימינה, או לא נוזו).
  - e. הוסף לקורדינטת y של ההדפסה מספר אקראי בין 1-ל-1.
  - f. אם הטקסט יצא מהמסך, מקם אותו במקום אקראי חדש.

התוכנית:

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <stdlib.h>

int main()
{
    int x, y;
    int i, temp;

    clrscr();
    randomize();

    x = random(80);
    y = random(24);

    for (i = 0; i <= 100; ++i)
    {
        textcolor(random(15)+1);
        gotoxy(x, y);
        cprintf("Hello, World");
        delay(100);
        gotoxy(x, y);
        textcolor(0);
        cprintf("Hello, World");
        temp = random(3) - 1;
        x = x + temp;
        temp = random(3) - 1;
        y = y + temp;
        if (x > 80 || x < 0) x = random(80);
        if (y > 24 || y < 0) y = random(24);
    }

    return 0;
}
```

EOF