

לולאות

קרון כליף

ביחידה זו נלמד:

3 סוגי לולאות: □

while ■

do-while ■

for ■

פקודות break ו- continue □

לולאות מקוננות □

לולאות - מוטיבציה

יהיו קטעי קוד שנרצה להריץ מספר פעמים, למשל הדפסת תו מסויים למסך 10 פעמים

```
#include <stdio.h>
void main()
{
    printf("*");
    printf("*");
    ...
    printf("*");
}
```

ע"י לולאה נוכל לחסוך בשורות קוד
■ לולאה – קטע קוד שנריץ יותר מפעם אחת

ע"י לולאה נוכל לעשות את התוכנית גמישה, כך שתריץ מספר כוכביות המשתנה מהרצה להרצה

לולאת while – דוגמת הדפסת 3 כוכביות למסך

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

```
void main()
```

```
{
```

```
    int i=0;
```

```
    while (i < 3)
```

```
    {
```

```
        printf("*");
```

```
        i++;
```

```
    }
```

```
    printf("\n");
```

```
}
```

Press any key to continue .

i = 0

* * *

כל סיבוב של הלולאה נקרא איטרציה

בכל סיבוב של הלולאה נקדם את i באחד כדי שהלולאה תסתיים לאחר 3 סיבובים. בלי הקידום הלולאה הייתה אינסופית והתוכנית לעולם לא הייתה מסתיימת..

לולאה שאמורה להתבצע מספר ידוע של פעמים תכיל את המרכיבים הבאים:

- ☐ אתחול
- ☐ בדיקת תנאי
- ☐ קידום

התפקיד של i הוא לשמור את מספר הכוכביות שכבר צוירו ולכן מאותחל ב- 0.

לולאת while – דוגמת הדפסת 3 כוכביות למסך

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

```
void main()
```

```
{
```

```
    int i=3;
```

```
    while (i > 0)
```

```
    {
```

```
        printf("*");
```

```
        i--;
```

```
    }
```

```
    printf("\n");
```

```
}
```

Press any key to continue .

i = 0

* * *

בקידום נקפיד לאמר שמקדמים את
הלולאה צעד אחד לקראת סופה, ולא
בהכרח שמקדמים את המשתנה i!

התפקיד של i הוא לשמור את מספר
הכוכביות שנותרו לציר ולכן מאותחל ב- 3

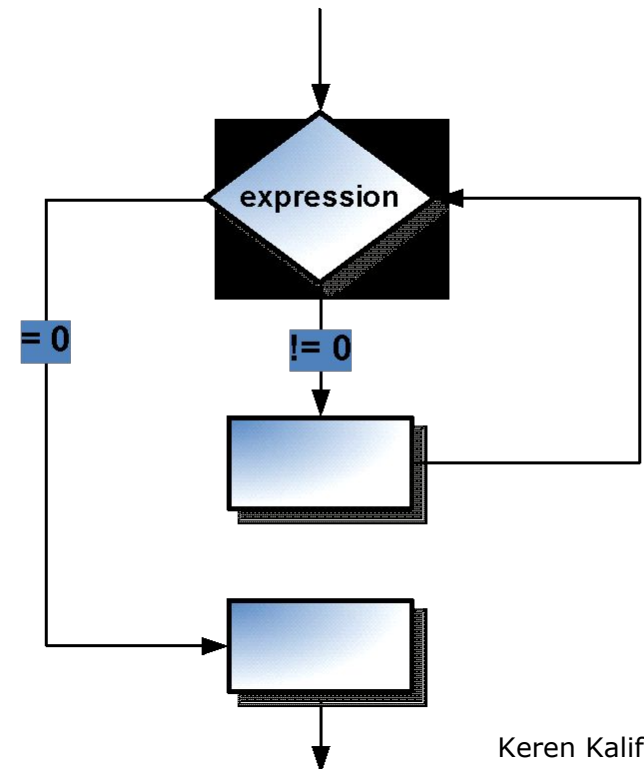
לולאת while

משפט מהצורה: □

```
while (<expression>)  
{  
    statement;  
    statement;  
    ...  
    statement;  
}
```

גם פה אין ; משום לא
מתבצעת פקודה (כמו ב- if)

כל עוד הביטוי נותן תוצאה שאינה 0, יבוצעו הפקודות שבגוף הלולאה



נשים לב, גם פה, שהקוד בין {} מוכנס טאב פנימה!

לולאת while – דוגמת הדפסת X כוכביות למסך

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

```
void main()
```

```
{
```

```
    int i=0;
```

```
    int numOfAstrix;
```

```
    printf("Please enter the number of astrix: ");
```

```
    scanf("%d", &numOfAstrix);
```

```
    while (i < numOfAstrix)
```

```
    {
```

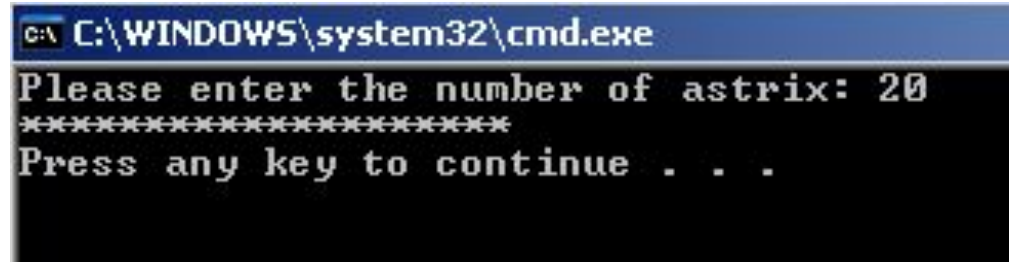
```
        printf("*");
```

```
        i++;
```

```
    }
```

```
    printf("\n");
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter the number of astrix: 20
*****
Press any key to continue . . .
```

לולאת while – חישוב ממוצע עד הכנסת -1

```
void main()  
{
```

```
    int num, counter=0, sum=0;
```

```
    printf("Please enter numbers, -1 to stop: ");  
    scanf("%d", &num);
```

```
    while (num != -1)  
    {
```

```
        sum += num;
```

```
        counter++;
```

```
        scanf("%d", &num);
```

```
    }
```

```
    printf("Average of the %d numbers is %f\n", counter,  
           (float)sum/counter);
```

```
}
```

C:\WINDOWS\system32\cmd.exe

```
Please enter numbers, -1 to stop: 6 3 4 8 -1  
Average of the 4 numbers is 5.250000  
Press any key to continue . . .
```

int: num	-1	1000
int: counter	2	1004
int: sum	9	1008

בדוגמא זו מספר האיטרציות אינו ידוע בשום שלב!

לולאת while – חישוב עצרת

```
Please enter a number: 4
4! = 24
Press any key to continue .
```

```
void main()
{
    int factorial=1, num, i=1;

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

    while (i <= num)
    {
        factorial *= i;
        i++;
    }
    printf("%d! = %d\n", num, factorial);
}
```

int: num	4	1000
int: factorial	24	1004
int: i	5	1008

לולאת while – חישוב סכום ספרותיו של מספר

```
void main()
{
    int num, sum=0;

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

    while (num != 0)
    {
        sum += num%10;
        num /= 10;
    }

    printf("Sum is %d\n", sum);
}
```

```
Please enter a number: 546
Sum is 15
Press any key to continue .
```

int: num	0	1000
int: sum	15	1004

גם בדוגמא זו לא יכולנו
לצפות את מספר האיטרציות

לולאה אינסופית

לולאה זו לא תסתיים לעולם מאחר והתנאי שלה תמיד מתקיים: □

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

```
void main()
```

```
{
```

```
    while (1)
```

```
    {
```

```
        printf("*");
```

```
    }
```

```
}
```

לולאה שלא תתבצע

גוף לולאה זו לא יבוצע אפילו פעם אחת! מדוע?

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

```
void main()
```

```
{
```

```
    int i = 10;
```

```
    while (i < 5)
```

```
    {
```

```
        printf("*");
```

```
        i++;
```

```
    }
```

```
}
```

וכאשר נרצה שהפעולה תתבצע לפחות פעם אחת..

```
void main()
```

```
{
```

```
    int number;=5;
```

כדי להמנע משכפול הפקודה, ניתן ערך dummy למשתנה על מנת שהלולאה תתבצע לפחות בפעם הראשונה

```
printf("Enter a positive number, negative to exit: ");
```

```
scanf("%d", &number);
```

```
while (number >= 0)
```

```
{
```

```
    printf("Enter a positive number, negative to exit: ");
```

```
    scanf("%d", &number);
```

```
}
```

```
printf("Thank you!\n");
```

```
}
```

```
Enter a positive number, negative to exit: 3
Enter a positive number, negative to exit: 5
Enter a positive number, negative to exit: -7
Thank you!
```

לולאת do-while

```
Enter a positive number, negative to exit: 3
Enter a positive number, negative to exit: 5
Enter a positive number, negative to exit: -7
Thank you!
```

```
void main()
{
    int number;

    do
    {
        printf("Please enter a positive number, negative to exit: ");
        scanf("%d", &number);
    } while (number >= 0);

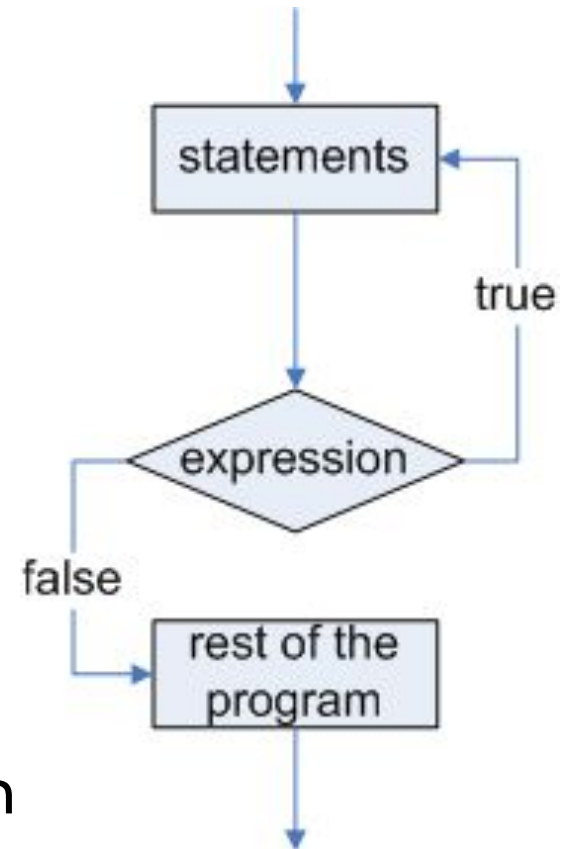
    printf("Thank you!\n");
}
```

באמצעות שימוש בלולאה זו אין
צורך להשתמש בערך dummy
באתחול או לשכפל קוד

לולאת do-while

```
do
{
    statement;
    statement;
    ...
    statement;
} while (<expression>);
```

משפט מהצורה: □



הפקודות בגוף הלולאה יבוצעו כל עוד התנאי מתקיים. גוף הלולאה יבוצע לפחות פעם אחת.

לולאת do-while

שימוש בתפריט

```
Choose one of the following options:
1- Squared number
2- draw line of starts
0- Exit
Enter your choice --> 1
Enter a number --> 3
3^2=9

Choose one of the following options:
1- Squared number
2- draw line of starts
0- Exit
Enter your choice --> 2
Enter num of starts --> 6
*****

Choose one of the following options:
1- Squared number
2- draw line of starts
0- Exit
Enter your choice --> 0

Goodbye!
Press any key to continue . . .
```

תפריט מורכב מ- 3 חלקים:

1. הצגת האפשרויות
2. switch לביצוע האפשרות הנבחרת
3. לולאה על-מנת לחזור על התפריט עד אשר המשתמש מבקש לצאת

מימוש תפריט

נשים לב כי ה- **break** מוציא אותנו
ממשפט הבקרה הפנימי ביותר, כלומר
בדוגמא זו מה- **switch** ולא

```
void main()
{
    int choice;
    int fContinue = 1;

    do
    {
        printf("Choose one of the following options:\n");
        printf("1- Squared number\n");
        printf("2- draw line of starts\n");
        printf("0- Exit\n");

        printf("Enter your choice --> ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
            {
                // here code of option 1
                break;
            }
            case 2:
            {
                // here code of option 2
                break;
            }
            case 0:
                fContinue = 0;
                break;
            default:
                printf("Invalid option");
                break;
        }

        printf("\n");
    } while (fContinue);

    printf("Goodbye!\n");
}
```

case 1:

```
{
    int num;

    printf("Enter a number --> ");
    scanf("%d", &num);

    printf("%d^2=%d\n", num, num * num);
    break;
}
```

case 2:

```
{
    int numOfStarts;

    printf("Enter num of starts --> ");
    scanf("%d", &numOfStarts);

    for (int i = 0; i < numOfStarts; i++)
        printf("*");
    printf("\n");
    break;
}
```

לולאת for – הדפסת כל המספרים

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

```
void main()
```

```
{
```

```
    int number, i;
```

```
    printf("Please enter a number: ");
```

```
    scanf("%d", &number);
```

```
    printf("All numbers from 1 to %d:\n", number);
```

```
    for ( i=1 ; i <= number ; i++ )
```

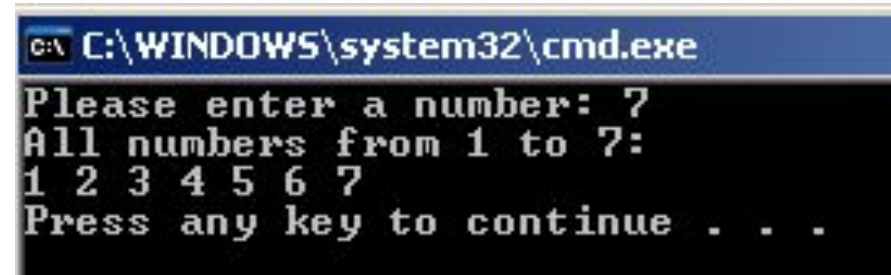
```
    {
```

```
        printf("%d ", i);
```

```
    }
```

```
    printf("\n");
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter a number: 7
All numbers from 1 to 7:
1 2 3 4 5 6 7
Press any key to continue . . .
```

int: number	3	1000
int: i	4	1004

1 2 3

איתחול ☐

כל עוד הביטוי נותן תוצאה שאינה 0: ☐

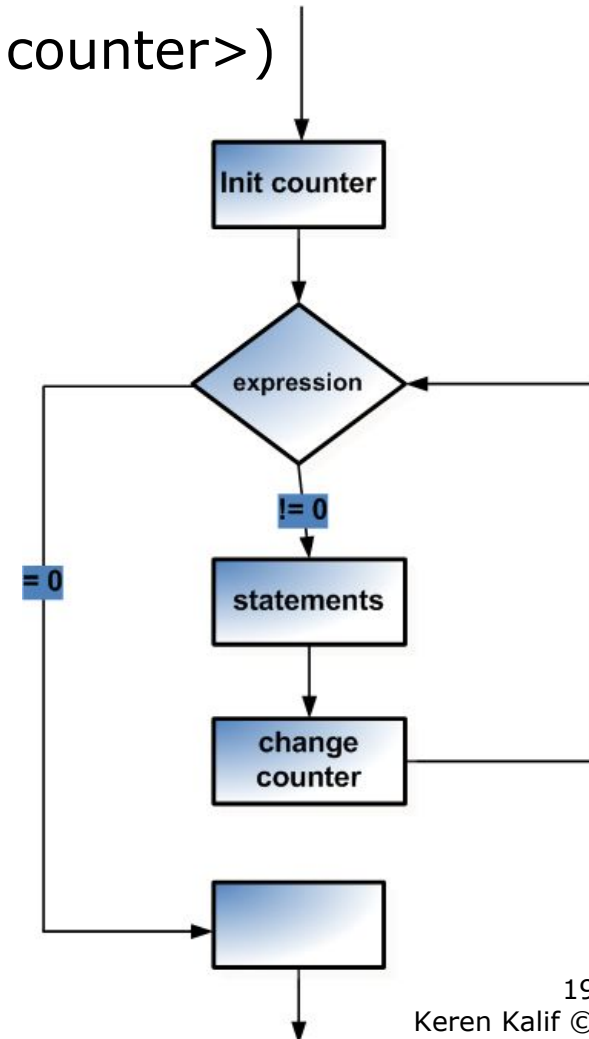
יבוצעו הפקודות שבגוף הלולאה ❖

קידום ❖

לולאת for

משפט מהצורה:

```
for (<init counter> ; <expression> ; <change counter>)  
{  
    statement;  
    statement;  
    ...  
    statement;  
}
```



נשים לב, גם פה, שהקוד בין {} מוכנס טאב פנימה!

לולאת for – הדפסת כל המספרים האי-הזוגיים

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

```
void main()
```

```
{
```

```
    int number, i;
```

```
    printf("Please enter a number: ");
```

```
    scanf("%d", &number);
```

```
    printf("All odd numbers from 1 to %d:\n", number);
```

```
    for ( i=1 ; i <= number ; i+=2 )
```

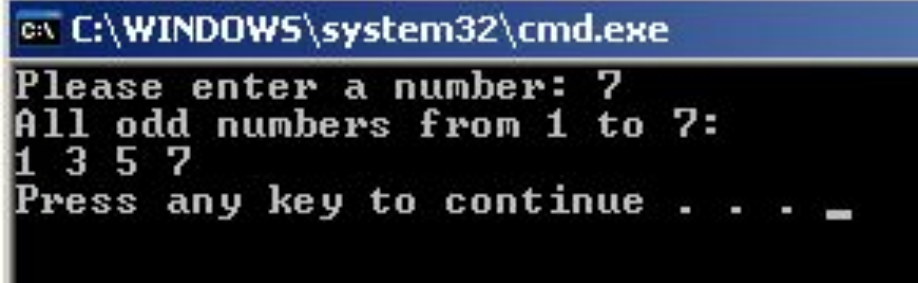
```
    {
```

```
        printf("%d ", i);
```

```
    }
```

```
    printf("\n");
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
```

```
Please enter a number: 7
```

```
All odd numbers from 1 to 7:
```

```
1 3 5 7
```

```
Press any key to continue . . . _
```

int: number	5	1000
int: i	7	1004

1 3 5

לולאת for – הדפסת כל המספרים בסדר יורד

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

```
void main()
```

```
{
```

```
    int number, i;
```

```
    printf("Please enter a number: ");
```

```
    scanf("%d", &number);
```

```
    printf("All numbers from %d to 1:\n", number);
```

```
    for ( i=number ; i >= 1 ; i-- )
```

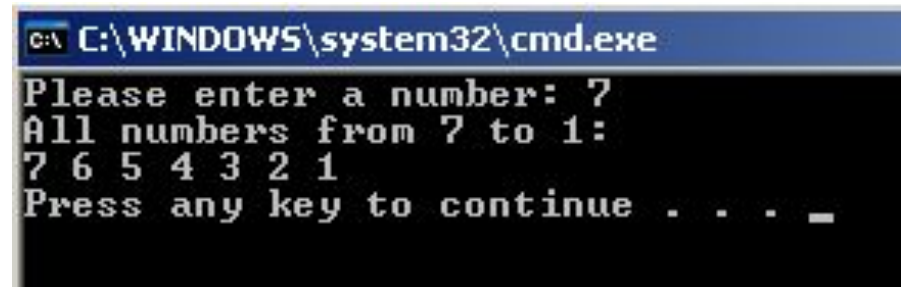
```
    {
```

```
        printf("%d ", i);
```

```
    }
```

```
    printf("\n");
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter a number: 7
All numbers from 7 to 1:
7 6 5 4 3 2 1
Press any key to continue . . . _
```

int: number	3	1000
int: i	0	1004

3 2 1

לולאת for – הדפסת חזקות של 2

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

```
void main()
```

```
{
```

```
    int number, i;
```

```
    printf("Please enter a number: ");
```

```
    scanf("%d", &number);
```

```
    printf("All 2 powers till %d:\n", number);
```

```
    for ( i=1 ; i <= number ; i *= 2 )
```

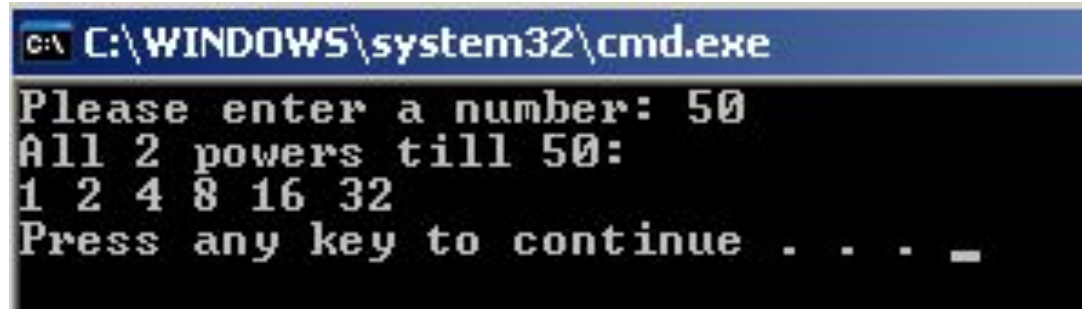
```
    {
```

```
        printf("%d ", i);
```

```
    }
```

```
    printf("\n");
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter a number: 50
All 2 powers till 50:
1 2 4 8 16 32
Press any key to continue . . . _
```

int: number	10	1000
int: i	16	1004

1 2 4 8

לולאת for – הדפסת חזקות של 2 (2 אינדקסים)

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

```
void main()
```

```
{
```

```
    int number, i, j;
```

```
    printf("Please enter a number: ");
```

```
    scanf("%d", &number);
```

```
    printf("All 2 powers till %d:\n", number);
```

```
    for ( i=1, j=0 ; i <= number ; i *= 2, j++ )
```

```
    {
```

```
        printf("2^%d = %d\n", j, i);
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
Please enter a number: 50
```

```
All 2 powers till 50:
```

```
2^0 = 1
```

```
2^1 = 2
```

```
2^2 = 4
```

```
2^3 = 8
```

```
2^4 = 16
```

```
2^5 = 32
```

```
Press any key to continue .
```

int: number	10	1000
int: i	16	1004
int: j	4	1008

2^0 = 1

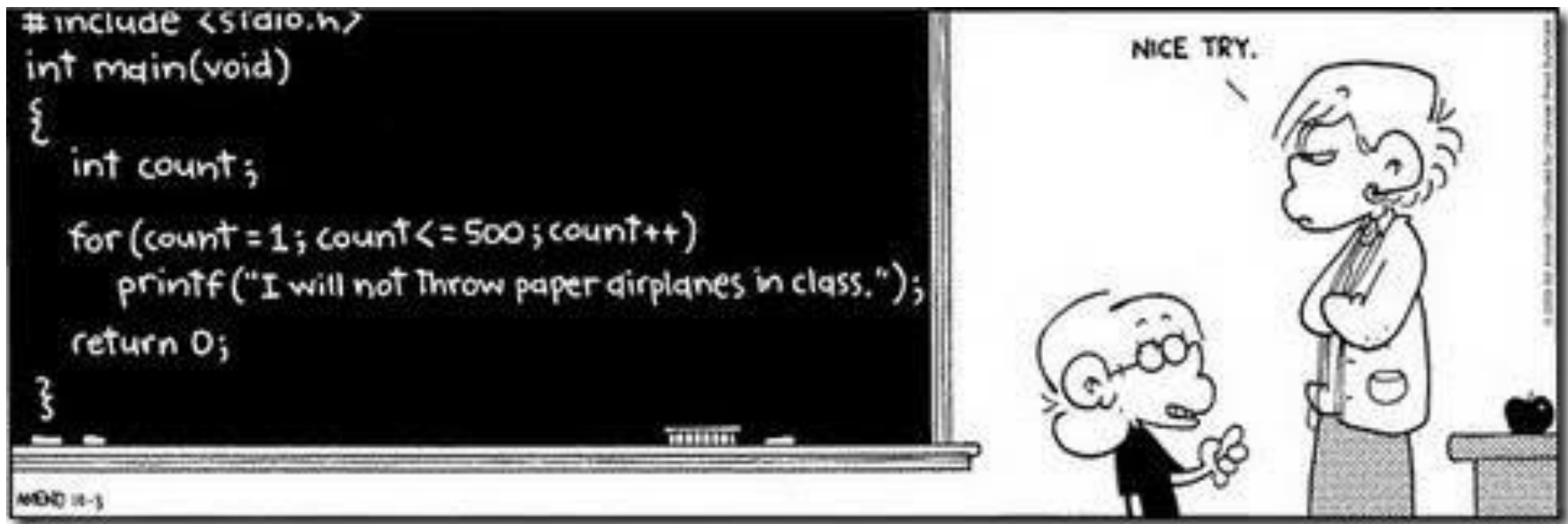
2^1 = 2

2^2 = 4

2^3 = 8

ניתן לשלב בלולאה כמה איתחולים
וכמה קידומים, מופרדים ע"י פסיק

שימוש יעיל ללולאת for



https://fbcdn-sphotos-c-a.akamaihd.net/hphotos-ak-ash2/t1/482216_430631353687511_1969402087_n.jpg

לולאת for ללא רכיב מסויים

```
int i=0;  
for ( ; i < 5 ; i++)  
    printf("*");
```

לולאה ללא איתחול

```
for (i=0 ; i < 5 ; )  
{  
    printf("*");  
    i++;  
}
```

לולאה ללא קידום

גם כאשר בלולאה אין רכיב מסויים יש
לכתוב את ה- ; בין חלקי הלולאה

```
for (i=0 ; ; i++)  
    printf("*");
```

לולאה ללא תנאי.
לא תסתיים לעולם!!

תרגום מלולאת while ללולאת for

```
int i;  
for ( i=0 ; i < 5 ; i++ )  
    printf("*");
```

ב- 2 הלולאות יש את 3 הרכיבים הבאים:

```
int i=0;  
while (i < 5)  
{  
    printf("*");  
    i++;  
}
```

☐ איתחול

☐ בדיקת תנאי

☐ קידום

שימוש בסוגי הלולאות השונים

□ כאשר צריך לכתוב קטע קוד שחוזר על עצמו, ניתן לבחור בכל אחת מהלולאות שראינו

□ יהיו מקרים בהם נעדיף לולאה מסויימת:

- אם מספר האיטרציות ידוע עם תחילת הלולאה, נעדיף לולאת for שכן האיתחול והקידום מובנים בתוכה
- אם מספר האיטרציות אינו ידוע עם תחילת הלולאה, נעדיף לולאה while, שכן אין צורך ב- counter שיש לקדמו לקראת סיום
- אם נרצה שהלולאה תבוצע לפחות פעם אחת, יש לשקול שימוש בלולאת do-while

יש להקפיד על שימוש בלולאה המתאימה בקוד אותו אתם

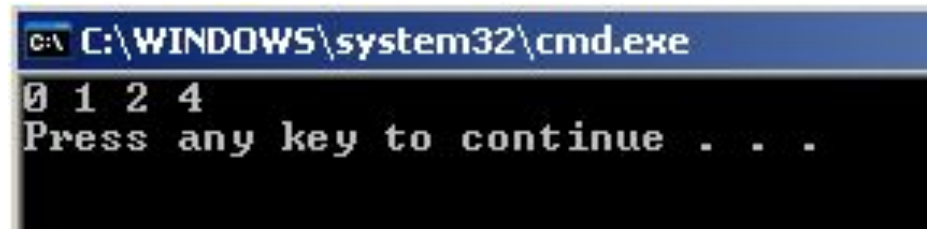
הפקודה continue

□ פקודה הגורמת לגוף הלולאה להפסיק ולהמשיך מיד לאיטרציה הבאה

```
void main()
{
    int i;
    for ( i=0; i < 5; i++ )
    {
        if (i == 3)
            continue;

        printf("%d ", i);
    }
    printf("\n");
}
```

הדפסת כל המספרים מ-0 עד 4 פרט ל-3:



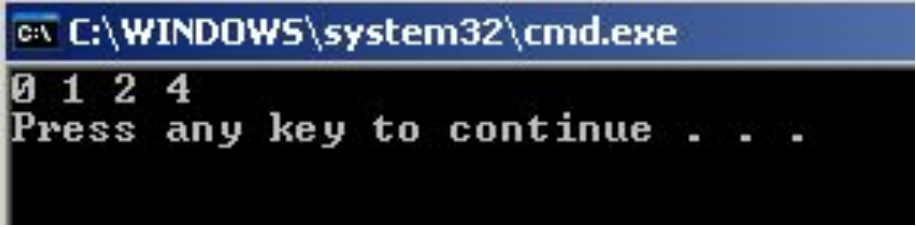
```
C:\WINDOWS\system32\cmd.exe
0 1 2 4
Press any key to continue . . .
```

הרצה: **0**

הפקודה continue - אפשר גם בלעדיה..

```
void main()
{
    int i=0;
    for ( i=0; i < 5 ; i++ )
    {
        if (i != 3)
            printf("%d ", i);
    }
    printf("\n");
}
```

הדפסת כל המספרים מ-0 עד 4 פרט ל-3:



```
C:\WINDOWS\system32\cmd.exe
0 1 2 4
Press any key to continue . . .
```

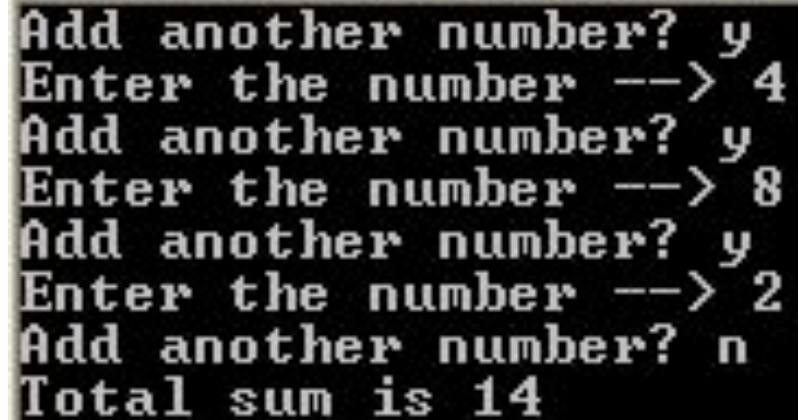
הפקודה: break

מפסיקה את הלולאה מיד

```
void main()
{
    int num, sum=0;
    char answer;

    while (1)
    {
        printf("Add another number? ");
        fflush();
        scanf("%c", &answer);
        if (answer == 'N' || answer == 'n')
            break;

        printf("Enter the number --> ");
        scanf("%d", &num);
        sum += num;
    {
        printf("Total sum is %d\n", sum);
    }
}
```



```
Add another number? y
Enter the number --> 4
Add another number? y
Enter the number --> 8
Add another number? y
Enter the number --> 2
Add another number? n
Total sum is 14
```

int: num	8	1000
int: sum	12	1004
char: answer	n	1008

דוגמה לשימוש ב- break

Algorithm of Success

```
while(noSuccess)
{
    tryAgain();

    if(Dead)
        break;
}
```

הפקודה break – אפשר גם בלעדיה..

```
void main()
{
    int num, sum=0, fContinue=1;
    char answer;

    while (fContinue) // fContinue == 1
    {
        printf("Add another number? ");
        fflush();
        scanf("%c", &answer);
        if (answer == 'N' || answer == 'n')
            fContinue = 0;
        else
        {
            printf("Enter the number --> ");
            scanf("%d", &num);
            sum += num;
        }
    }
    printf("Total sum is %d\n", sum);
}
```

```
Add another number? y
Enter the number --> 4
Add another number? y
Enter the number --> 8
Add another number? y
Enter the number --> 2
Add another number? n
Total sum is 14
```

int: num	8	1000
int: sum	12	1004
char: answer	n	1008
int: fContinue	0	1012

הפקודה break – בלעדיה אפשר גם כך..

```
void main()
{
    int num, sum=0, fStop=0;
    char answer;

    while (!fStop) // fStop == 0
    {
        printf("Add another number? ");
        fflush();
        scanf("%c", &answer);
        if (answer == 'N' || answer == 'n')
            fStop = 1;
        else
        {
            printf("Enter the number --> ");
            scanf("%d", &num);
            sum += num;
        }
    }
    printf("Total sum is %d\n", sum);
}
```

```
Add another number? y
Enter the number --> 4
Add another number? y
Enter the number --> 8
Add another number? y
Enter the number --> 2
Add another number? n
Total sum is 14
```

int: num	8	1000
int: sum	12	1004
char: answer	n	1008
int: fStop	1	1012

לולאות מקוננות – הדפסת ריבוע

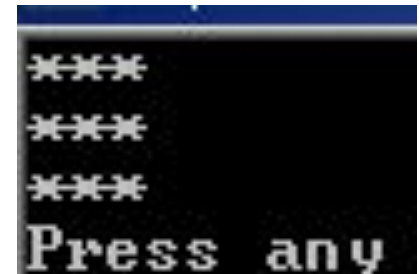
```
#include <stdio.h>
```

```
#define SIZE 3
```

```
void main()
```

```
{  
    int i, j;  
  
    for ( i=0 ; i < SIZE ; i++ )  
    {  
        for ( j=0 ; j < SIZE ; j++)  
            printf("*");  
        printf("\n");  
    }  
}
```

```
* * *  
* * *  
* * *
```



int: i	3	1000
int: j	3	1004

יכולנו גם להתחיל $i=1$ ואז לרוץ עם התנאי $i \leq \text{SIZE}$, וכנ"ל עם j

לולאות מקוננות – הדפסת מלבן

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

```
void main()
```

```
{
```

```
    int i, j, rows, cols;
```

```
    printf("Please enter number of rows and columns: ");
```

```
    scanf("%d %d", &rows, &cols);
```

```
    for (i=0 ; i < rows ; i++)
```

```
    {
```

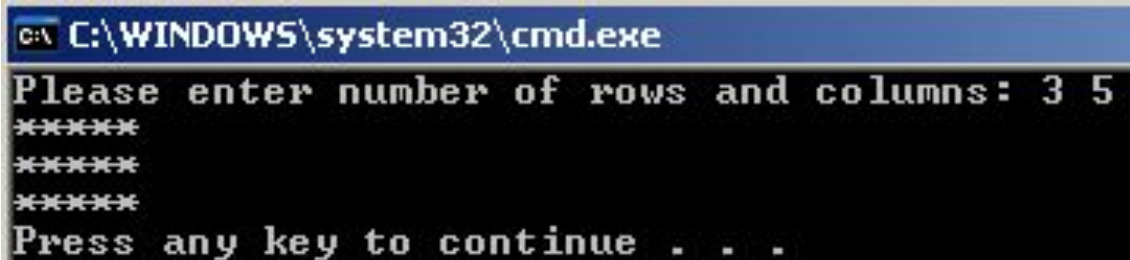
```
        for (j=0 ; j < cols ; j++)
```

```
            printf("*");
```

```
        printf("\n");
```

```
    }
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe
Please enter number of rows and columns: 3 5
*****
*****
*****
Press any key to continue . . .
```

יכולנו גם להתחיל $i=1$ ואז לרוץ עם התנאי $i \leq \text{SIZE}$, וכנ"ל עם i

לולאות מקוננות – הדפסת משולש

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

```
void main()
{
```

```
    int i, j, base;
```

```
    printf("Please enter the base of the triangle: ");
    scanf("%d", &base);
```

```
    for ( i=1 ; i <= base ; i++ )
    {
        for ( j=1 ; j <= i ; j++ )
            printf("*");
        printf("\n");
    }
```

```
}
```

```
C:\WINDOWS\system32\cmd.exe
Please enter the base of the triangle: 5
*
**
***
****
*****
Press any key to continue . . .
```

# כוכביות בשורה	# שורה
1	1
2	2
3	3
4	4
5	5

```
*
* *
* * *
```

int: i	4	1000
int: j	4	1004
int: base	3	1008

האם יכולנו להתחיל את אחד האינדקסים מ-

לולאות מקוננות – הדפסת משולש מיושר

לימין

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

```
void main()  
{
```

```
    int i, j, base;
```

```
    printf("Please enter the base of the triangle: ");  
    scanf("%d", &base);
```

```
    for (i=1 ; i <= base ; i++)  
    {
```

```
        for (j=1 ; j <= ??? base-i . )  
            printf(" ");
```

```
        for (j=1 ; j <= i ; j++)  
            printf("*");
```

```
        printf("\n");
```

```
    }
```

```
}
```

```
C:\WINDOWS\system32\cmd.exe  
Please enter the base of the triangle: 4  
 *  
 **  
 ***  
 ****  
Press any key to continue . .
```

# שורה	# רווחים בשורה	# כוכביות בשורה
1	3	1
2	2	2
3	1	3
4	0	4

מהטבלה ברור כי מספר הרווחים בכל שורה הוא מספר הכוכביות המקסימלי (base) פחות מספר הכוכביות בשורה (i)

לולאות מקוננות – הדפסת משולש מיושר לימין – אפשר גם כך..

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

```
void main()  
{
```

```
    int i, j, k, base;
```

```
    printf("Please enter the base of the triangle: ");  
    scanf("%d", &base);
```

```
    for (i=1, k=base-1 ; i <= base ; i++ , k-- )  
    {
```

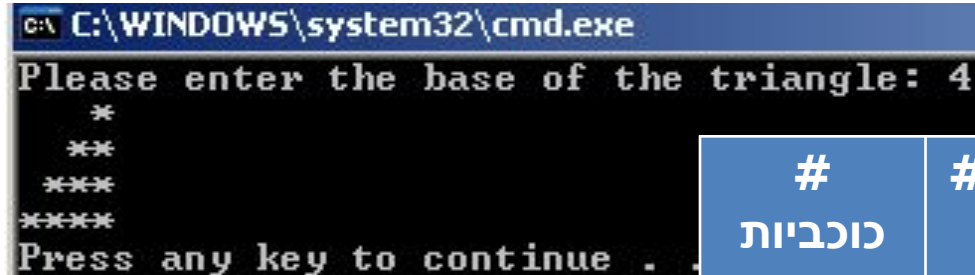
```
        for (j=1 ; j <= ??? k ; j++)  
            printf(" ");
```

```
        for (j=1 ; j <= i ; j++)  
            printf("*");
```

```
        printf("\n");
```

```
    }
```

```
}
```



```
C:\WINDOWS\system32\cmd.exe  
Please enter the base of the triangle: 4  
 *  
  *  
   *  
    *  
Press any key to continue . .
```

# שורה	# רווחים בשורה	# כוכביות בשורה
1	3	1
2	2	2
3	1	3
4	0	4

מהטבלה ניתן לראות כי מספר הרווחים
קטנים ב-1 בכל שורה, והערך הראשוני
הוא $base-1$

הוספת ספרה מימין למספר

- כאשר רוצים להוסיף ספרה מימין למספר, כלומר להכניס ספרת אחדות חדשה, יש לבצע את הפעולות הבאות:
 - הכפל את המספר פי 10 (כדי לייצר מקום לספרת האחדות החדשה)
 - הוסיף את הספרה החדשה

- דוגמא: עבור המספר 65 נרצה להוסיף את הספרה 3 מימין, כלומר כדי לייצר את מספר 653:
 - $650 = 10 * 65$
 - $653 = 3 + 650$

דוגמה: יצירת המספר ההופכי

- למשל, עבור 123 יש לייצר את המספר 321
- במקרה זה צריך לבודד את כל אחת מהספרות ולהוסיפן מימין למספר שאותו מייצרים ולכן נשתמש בפעולות הבאות:
 - "תן ספרה ימנית" (מודולו ב- 10)
 - "קצץ ספרה ימנית" (חילוק ב- 10)
- ניתן להשתמש גם בפעולות חשבון בסיסיות (כפל, חיבור וכד')

תרגיל: יצירת המספר ההופכי

1. הגדר $\text{newNum} = 0$
2. קלוט מספר לתוך num
3. כל עוד $\text{num} > 0$:
 - (a) הכפל את הערך של newNum פי 10
 - (b) תן ספרה ימנית מ- num והוסף אותה ל- newNum
 - (c) קצץ ספרה ימנית מ- num
4. הצג את newNum

הרצה יבשה:

$\text{newNum} = 000$

$\text{num} = 023$

נשים לב שהמספר המקורי num נהרס, ואם
נרצה לעשות בו שימוש בסוף לא נוכל

תרגום לשפת C

```
void main()
{
    int num, newNum=0;

    printf("Enter a number: ");
    scanf("%d", &num);

    while (num > 0)
    {
        newNum *= 10;
        newNum += num%10;
        num /= 10;
    }

    printf("Rev num is %d\n", newNum);
}
```

1. הגדר $\text{newNum} = 0$

2. קלוט מספר לתוך num

3. כל עוד $\text{num} > 0$:

(a) הכפל את הערך של newNum פי 10

(b) תן ספרה ימנית מ- num והוסף אותה ל- newNum

(c) קצץ ספרה ימנית מ- num

4. הצג את newNum

שמירה על הערך המקורי

```
void main()
{
    int num, newNum , temp;
```

```
    printf("Enter a number: ");
    scanf("%d", &num);
```

```
    temp = num;
    while ( temp > 0 )
    {
        newNum *= 10;
        newNum += n temp%10;
        n temp /= 10;
    }
```

ערכו של num פה הוא 0..

```
    printf("Rev of %d is %d\n", num, newNum);
}
```

הוספת ספרות משמאל למספר

□ כאשר מוסיפים ספרה משמאלו של מספר יש לקחת בחשבון איזה מיקום היא תופסת: אחדות / עשרות / מאות וכד'

□ דוגמא: אם רוצים למספר 47 להוסיף את הספרה 3 משמאלו, למעשה יש להוסיף את הערך 300

$$347 = 300 + 47 \quad \blacksquare$$

■ אם נרצה להוסיף ספרה נוספת משמאל, היא כבר תהייה במיקום של האלפים

הוספת ספרות משמאל למספר - דוגמא

□ נייצר את המספר 724 תוך כדי קבלת ספרותיו
מהאחדות למאות: 4 2 7

הרצה יבשה:

המספר: 720

□ בהתחלה המספר שבונים יהיה 0

□ נוסיף למספר את הערך 4

$$1 * 4 = 4 \quad \blacksquare$$

□ נוסיף למספר את הערך 20

$$10 * 2 = 20 \quad \blacksquare$$

□ נוסיף למספר את הערך 700

$$100 * 7 = 700 \quad \blacksquare$$

□ המספר המתקבל הוא 724

ניתן לזהות את החוקיות שהוספת ספרה
משמאל היא למעשה כפולה כלשהי של
10, שגדלה פי 10 בכל סיבוב

שימוש באלגוריתם

קלוט ספרות של מספר כאשר הספרה הראשונה הינה ספרת האחדות, השניה עשרות וכו' עד אשר מוקלד הערך 1- . ייצר משתנה מטיפוס int שיכיל את המספר שספרותיו הוקלדו

הרצה יבשה:

```
newNum = 638  
digit = -1  
location = 1000
```

1. הגדר $\text{newNum} = 0$

2. הגדר digit

3. הגדר את המשתנה $\text{location} = 1$

4. קלוט ספרה לתוך digit

5. כל עוד $\text{digit} \neq -1$:

(a) הכפל את הספרה ב- location והוסף את המכפלה ל- newNum

(b) הכפל את location פי 10

(c) קלוט ספרה לתוך digit

6. הצג את newNum

תרגום לשפת C

```
void main()
{
    int digit, newNum=0, location=1;

    printf("Enter a digit: ");
    scanf("%d", &digit);

    while (digit != -1)
    {
        newNum += digit*location;
        location *= 10;

        printf("Enter a digit: ");
        scanf("%d", &digit);
    }

    printf("The num is %d\n", newNum);
}
```

1. הגדר $\text{newNum} = 0$

2. הגדר digit

3. הגדר את המשתנה $\text{location}=1$

4. קלוט ספרה לתוך digit

5. כל עוד $\text{digit} \neq -1$:

(a) הכפל את הספרה ב- location והוסף את המכפלה ל- newNum

(b) הכפל את location פי 10

(c) קלוט ספרה לתוך digit

6. הצג את newNum

```
void main()
```

```
{
```

```
    int digit, newNum=0, location=1;
```

```
    printf("Enter a digit: ");  
    scanf("%d", &digit);
```

```
    while (digit != -1)
```

```
    {
```

```
        newNum += digit*location;  
        location *= 10;
```

```
        printf("Enter a digit: ");  
        scanf("%d", &digit);
```

```
    }
```

```
    printf("The num is %d\n", newNum);
```

```
{
```

גרסה משופרת

```
void main()
```

```
{
```

```
    int digit, newNum=0, location=1;
```

```
    int fContinue = 1;
```

```
    while (fContinue)
```

```
    {
```

```
        printf("Enter a digit: ");  
        scanf("%d", &digit);
```

```
        if (digit == -1)
```

```
            fContinue = 0;
```

```
        else
```

```
        {
```

```
            newNum += digit*location;  
            location *= 10;
```

```
        }
```

```
    }
```

```
    printf("The num is %d\n", newNum);
```

```
{
```


והידה לסיום!

This problem can be solved by pre-school children in five to ten minutes, by programmers in an hour and by people with higher education... well, check it yourself!

$$8809 = 6$$

$$7111 = 0$$

$$2172 = 0$$

$$6666 = 4$$

$$1111 = 0$$

$$3213 = 0$$

$$7662 = 2$$

$$9313 = 1$$

$$0000 = 4$$

$$2222 = 0$$

$$3333 = 0$$

$$5555 = 0$$

$$8193 = 3$$

$$8096 = 5$$

$$1012 = 1$$

$$7777 = 0$$

$$9999 = 4$$

$$7756 = 1$$

$$6855 = 3$$

$$9881 = 5$$

$$5531 = 0$$

$$2581 = ???$$

SCHOOL.FAILBLOG.ORG

הכנה לקראת שיעורי הבית מהיום והלאה



http://rlv.zcache.ca/programmers_do_it_all_night_greeting_card-r603fa91b18e1473192a9e0b8c5e2cd8d_xvuak_8byvr_324.jpg

ביחידה זו למדנו:

3 סוגי לולאות: □

while ■

do-while ■

for ■

פקודות break ו- continue □

לולאות מקוננות □