**לולאת FOR בשפת C**

**סיכום שיעור מקוון מתאריך 19/5/2020**

תחביר הלולאה :

פקודות לפני …

**1**

**2**

**3**

**4**

**5**

for ( ערך התחלתי ; תנאי להמשך הלולאה ; שינוי )

{

גוף הלולאה ;

}

פקודות אחרי …

הסבר :  
ערך התחלתי – מתבצע ראשון ובאופן חד-פעמי

תנאי – נבדק לאחר מכן : אם הוא אמת, המחשב יתחיל בביצוע גוף הלולאה. אם לא, המחשב ידלג להמשך התכנית.

שינוי – יתבצע תמיד מיד אחרי סיום ביצוע גוף הלולאה, ומשם שוב נחזור לתנאי.

דוגמא :

#include <stdio.h>

int main()

{

printf("Hi\n");

for (a=3 ; a>0 ; a--)

{

printf("\*");

}

printf("\nBye");

}

|  |  |  |
| --- | --- | --- |
|  | 3 |  |
| מחזור | a | מסך |
| 1 | 2 | \* |
| 2 | 1 | \* |
| 3 | 0 | \* |

**משימה 1**

הדפס למסך את כל המספרים הדו-ספרתיים, המתחלקים ל-5.

**פתרון :**

דו-ספרתיים => לולאת מונה הרצה מ-10 עד 99.

גרסה א' :

#include <stdio.h>

int main()

{

int x ;

for (x=10 ; x<=99 ; x++)

{

if (x%5==0)

printf("%d\t",x);

}

}

גרסה ב' :

#include <stdio.h>

int main()

{

int x ;

for (x=10 ; x<=99 ; x+=5)

{

printf("%d\t",x);

}

}



**משימה 2**

קלוט מספר שלם והדפס למסך את כל המחלקים שלו (לא כולל 1 ועצמו).

לדוגמא, אם קלטנו : number = 36, למסך יודפס : 2 , 3 , 4 , 6 , 9 , 12 , 18

**פתרון :**

רצוי להשתמש בלולאה הרצה מ-2 ועד למחצית המספר (כולל).

#include <stdio.h>

int main()

{

int num,i ;

printf("Enter a number : ");

scanf("%d",&num);

for (i=2 ; i<=num/2 ; i++)

if (num%i==0)

printf("%d ",i) ;

// (num%i==0) ? printf("%d ",i) : printf("");

}



**משימה 3**

"מספר ראשוני" - מספר בלתי פריק, כלומר מספר המתחלק ל-1 ולעצמו בלבד.

קלוט מספר שלם ובדוק האם הוא ראשוני :  
אם כן – מדפיסה "YES"  
אם לא – מדפיסה "NO"

**פתרון :**

343 : 2 = יש שארית

343 : 3 = יש

343 : 4 = יש

…

343 : 171 = יש

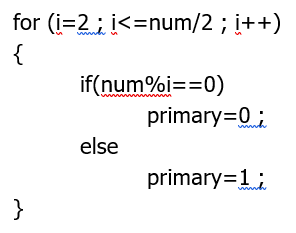
אם נחלק את המספר הנבדק לכל אחד מהמספרים שבטווח שבין 2 למחציתו, ותמיד נקבל שארית. סימן שהוא ראשוני.

25 : 2 = יש

25 ; 3 = יש  
25 : 4 = יש

25 : 5 = אין

במידה ולפחות אחת החלוקות היא ללא שארית (שארית=0), סימן שהמספר אינו ראשוני ואין טעם להמשיך ולבדוק את יתר החלוקות.

לכאורה, נראה כי זה הפתרון המתאים :

אך בזמן ההרצה, מסתגר שאין זה כך.

נאתר את הבעיה עבור : num=25

|  |  |  |
| --- | --- | --- |
|  |  |  |
| i | num%i | primary |
| 2 | 25%2=1 | 1 |
| 3 | 25%3=1 | 1 |
| 4 | 25%4=1 | 1 |
| 5 | 25%5=0 | 0 |
| 6 | 25%6=1 | 1 |
| 7 |  | 1 |
| 8 |  | 1 |
| 9 |  | 1 |
| 1 |  | 1 |
| 12 | 25%12=1 | 1 |

הפתרון הוא לעצור ברגע שקיבלנו חלוקה ללא שארית. כמו כן, את ההדפסה נבצע מחוץ ללולאה, כדי לקבל פלט אחד ויחיד :

#include <stdio.h>

int main()

{

int num,i, primary=1 ;

printf("Enter a number : ");

scanf("%d",&num);

for (i=2 ; i<=num/2 && primary==1 ; i++)

{

if(num%i==0)

primary=0 ;

}

if(primary==1)

printf("YES");

else

printf("NO");

}

פתרון נוסף :

#include <stdio.h>

int main()

{

int num,i, primary=0;

printf("Enter a number : ");

scanf("%d",&num);

for (i=2 ; i<=num/2 ; i++)

{

if(num%i==0)

primary++ ;

}

if(primary==0)

printf("YES");

else

printf("NO");

}





**משימה 4**

קלוט מספר שלם למשתנה number וחשב את המספר ההפוך למספר number והצב אותו במשתנה newNumber.  
לדוגמא, אם קלטנו : number = 1984, המספר החדש יהיה : newNumber = 4891.

רמז :

* כדי להפוך מספר, יש תחילה לפרק אותו לספרות.

number = 1984 = 100 + 900 + 80 + 4 = 1X1000 + 9X100 + 8X10 + 4X1

newNumber = 4891 = 4X1000 + 8X100 + 9X10 + 1X1

* כדי לפרק מספר לספרות, יש לחלק אותו ל-10 באמצעות % וגם /, בתוך לולאה, עד אשר לא נשאר דבר מהמספר. כלומר, המספר מתאפס.
* את תהליך הפירוק של המספר number ותהליך הבנייה של המספר newNumber מבצעים באותה הלולאה !!!

**פתרון :**

newNum=0

1984/10 1984%10=4 newNum=0 + 4

198/10 198%10=8 newNum=48 = 40 + 8

19/10 19%10=9 newNum=489 = 480 + 9

1/10 1%10=1 newNum=4891 = 4890 + 1

0

25/10 25%10=5

2/10 2%10=2

#include <stdio.h>

int main()

{

int num, digit, newNum=0;

printf("Enter a number : ");

scanf("%d",&num) ;

for ( ; num!=0 ; num/=10)

{

digit = num%10 ;

newNum = newNum\*10 + digit ;

}

printf("newNum = %d",newNum);

}

**משימה 5**

קלוט שני מספרים שלמים ובדוק האם הם הפוכים האחד לשני :  
אם כן – הדפס "YES"

אם לא – הדפס "NO"

לדוגמא, אם קלטנו : a=123 , b=321 , יודפס למסך "YES".

הערה : כדאי להיעזר בפתרון של שאלה 4.

**פתרון :**

#include <stdio.h>

int main()

{

int num1, num2, digit, newNum=0;

printf("Enter the first number : ");

scanf("%d",&num1) ;

printf("Enter the second number : ");

scanf("%d",&num2) ;

for ( ; num1!=0 ; num1/=10)

{

digit = num1%10 ;

newNum = newNum\*10 + digit ;

}

if (newNum == num2)

printf("YES");

else

printf("NO");

}

**הערות לגביי לולאת FOR**

**1) אתחול ושינוי שלא בשורת ה-for**

לעיתים יותר נוח לאתחל ו/או לשנות את משתנה הלולאה שלא במקום הרגיל.

לדוגמא, קלוט סדרת מספרים וסכם אותם, עד אשר נקלט המספר 0.

#include <stdio.h>

int main()

{

int mispar, sum=0 ;

printf("Enter the first number : ");

scanf("%d",&mispar) ;

for ( ; mispar!=0 ; )

{

sum += mispar ;

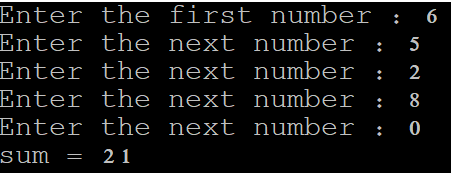
printf("Enter the next number : ");

scanf("%d",&mispar);

}

printf("sum = %d",sum);

}



**2) לולאה ריקה, סגורה**

יש מקרים מיוחדים בהם ניתן לסגור את לולאת ה-for בסוף השורה שלה (סגורה). במקרה זה ללולאה אין גוף (ריקה).

לדוגמא, מצא את המספר התלת ספרתי הראשון המתחלק ל-13.

**פתרון א' :**

#include <stdio.h>

int main()

{

int x, found=0 ;

for (x=100 ; x<=999 && found==0 ; x++)

{

if (x%13==0)

found=1 ;

}

if (found==1)

printf("%d",x-1);

else

printf("Not found");

}

**פתרון ב' : לולאה סגורה**

#include <stdio.h>

int main()

{

int x ;

for (x=100 ; x<=999 && x%13!=0 ; x++) **;**

if (x<=999)

printf("%d",x);

else

printf("Not found");

}

**3) אתחול ושינוי מרובים**

ניתן לאתחל מספר משתנים בשורת ה-for.  
כמו כן, ניתן לשנות מספר משתנים בשורת ה-for.

כל אלה אפשריים באמצעות פסיק מפריד.

לדוגמא :

#include <stdio.h>

int main()

{

int x,y ;

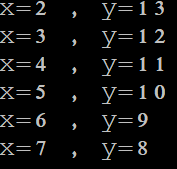
for (x=2,y=13 ; x<y ; x++,y--)

{

printf("x=%d , y=%d\n",x,y);

}

}



**4) שימוש ב-break**

הפקודה break "שוברת החוצה" מתוך הפקודה שבתוכה היא רשומה.

השתמשנו בעבר בפקודת break בהקשר של שבירת הפקודה switch, ואילו עכשיו נראה השימוש בה בהקשר של לולאות.

הפקודה break יודעת לשבור, כלומר להפסיק לאלתר את פעולת הלולאה.

לדוגמא, קלוט 5 מספרים חיוביים וסכם אותם. במידה ונקלט מספר שלילי, עצור את התהליך.

#include <stdio.h>

int main()

{

int x ,num ,sum=0 ;

for (x=1 ; x<=5 ; x++)

{

printf("Enter a number : ");

scanf("%d",&num);

if (num<0)

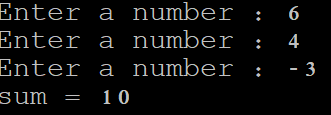
break ;

sum+=num ;

}

printf("sum = %d",sum);

}



דוגמא נוספת : מימוש הפתרון של מספר ראשוני, באמצעות break

#include <stdio.h>

int main()

{

int num,i ;

printf("Enter a number : ");

scanf("%d",&num);

for (i=2 ; i<=num/2 ; i++)

{

if(num%i==0)

break ;

}

if(num%i==0)

printf("NO");

else

printf("YES");

}

**5) שימוש ב-continue**

הפקודה continue יכול להופיע אך ורק בתוך לולאות ותפקידה לדלג למחזור הבא.

לדוגמא, קלוט 5 מספרים וסכם אותם. במידה ונקלט מספר שלילי, התעלם ממנו ועבור למספר הבא.

#include <stdio.h>

int main()

{

int x ,num ,sum=0 ;

for (x=1 ; x<=5 ; x++)

{

printf("Enter a number : ");

scanf("%d",&num);

if (num<0)

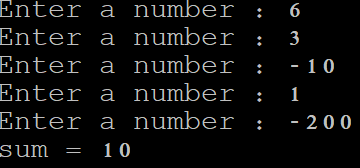
continue ;

sum+=num ;

}

printf("sum = %d",sum);

}



**משימה 6**

קלוט מספר שלם ובדוק האם סכום המחלקים שלו (כולל 1, לא כולל עצמו) שווה למספר עצמו ("מספר מושלם") :

אם כן – הדפס "YES"

אם לא – הדפס "NO"

לדוגמא :

* num = 12, יודפס : NO  
  הסבר : 1+2+3+4+6=16 ≠ 12
* num = 6, יודפס : YES  
  הסבר : 1+2+3=6
* num = 28, יודפס : YES  
  הסבר : 1+2+4+7+14=28

פתרון :

#include <stdio.h>

int main()

{

int num,x,sum=0;

printf("plese enter a num :");

scanf("%d",&num);

for(x=1 ; x<num ; x++)

{

if (num%x==0)

sum= sum+x;

}

if (sum!=num)

printf("no");

else

printf("yes");

}

\* הערה : מספיק לרוץ עד למחצית המספר (כולל).

נסביר את הפתרון בעזרת טבלת מעקב, עבור num=6 :

|  |  |  |
| --- | --- | --- |
| 1 |  | 0 |
| **x** | **num%x** | **sum** |
| 2 | 6%1=0 | 0+1=1 |
| 3 | 6%2=0 | 1+2=3 |
| 4 | 6%3=0 | 3+3=6 |
| 5 | 6%4=2 |  |
| 6 | 6%5=1 |  |

**משימה 7** – תצורף למטלה 6

כתוב תוכנית הקולטת זוג מספרים שלמים וחיוביים ומחשבת את המחלק המשותף הקטן והגדול ביותר (לא כולל 1).

לדוגמא, עבור המספרים : 24 , 60 יודפסו לפלט : 2 , 12.  
לדוגמא, עבור המספרים : 24 , 9 יודפסו לפלט : 3 , 3.  
לדוגמא, עבור המספרים : 33 , 25 תודפס ההודעה : "לא קיים מחלק שכזה".

רמז :

* יש להיעזר במשתנה שייצג את המחלק הקטן ביותר ומשתנה נוסף שייצג את המחלק הגדול ביותר.
* יש לאתחל את המשתנים הללו ב-0, עבור המקרה השלישי.