**תרגול לקראת בחינת סמסטר**

**מתוך :** [**פרויקט אוילר**](https://projecteuler.net/about)

**שאלה 4**

A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is 9009 = 91 × 99.

Find the largest palindrome made from the product of two 3-digit numbers.

מספר palindromic נקרא את אותו הדבר בשני הכיוונים. הפלינדרום הגדול ביותר העשוי מתוצר של שני מספרים דו ספרתיים הוא 9009 = 91 × 99.

מצא את הפלינדרום הגדול ביותר המיוצר מתוצר של שני מספרים תלת ספרתיים.

**פתרון :**

* מכפלה של מספרים דו-ספרתיים, 10 עד 99 כפול 10 עד 99. יש לרוץ הפוך, מהמכפלה הכי גדולה, אחורה. כל מכפלה יש לבדוק האם התוצאה שלה היא מספר פלינדרומי :
  + אם כן – לעצור ולהדפיס.
  + אם לא – להמשיך הלאה, עד שמוצאים או עד שמסיימים את הלולאה.
* כדי לבדוק אם המכפלה היא פלינדרומית, נשלח אותה לפונקצית עזר :
  + אם כן – היא תחזר 1.
  + אם לא - היא תחזיר 0.
* איך בודקים ? ניקח את המספר הנבדק וניצור ממנו מספר חדש הפוך, ונגלה ששניהם שווים האחד לשני, הרי שהוא פלינדרום.

num = 122

new num = 221

221 ≠ 122 => no return 0

num = 9009

new num = 9009

9009 = 9009 => yes return 1

#include <stdio.h>

**int isPalindrom(int r)**

{

int flip = 0, tmp=r;

while (r)

{

flip = flip\*10 + (r%10);

r /= 10;

}

return flip == tmp ;

/\*if (flip==tmp)

return 1 ;

else

return 0 ; \*/

}

**int main()**

{

int x,y,res, pal=0, max=0 ;

printf("%d\n",999\*999);

for (x=999 ; x>=100 ; x--)

for (y=999 ; y>=100 ; y--)

{

res=x\*y ;

pal = isPalindrom(res) ;

if (pal==1 && res>max)

max=res ;

}

printf("res = %d",max);

}

**שאלה 5**

2520 is the smallest number that can be divided by each of the numbers from 1 to 10 without any remainder.

What is the smallest positive number that is *evenly divisible* by all of the numbers 2520 הוא המספר הקטן ביותר שניתן לחלק על ידי כל אחד מהמספרים מ- 1 עד 10 ללא כל שארית.

מה המספר החיובי הקטן ביותר שמתחלק באופן שווה בין כל המספרים מ -1 עד 20?from 1 to 20?

**פתרון :**

* המספר שאנו מחפשים חייב להיות כפולות של 20, ולכן נתחיל לחפש מ-20 ונתקדם בקפיצות של 20 = הלולאה החיצונית.
* בתוך הלולאה החיצונית, נבדוק כל מספר ע"י חלוק שלו לכל המספרים מ-2 (לחלק ל-1 זה מיותר), עד 19 (כי בטוח הוא מתחלק ל-20) = לולאה פנימית.
* ברגע שנמצא מספר שמתחלק לכל אלה (2-19), נעצור ונדפיס אותו.

#include <stdio.h>

**int checkX(int num)**

{

int i ;

for (i=19 ; i>=2 ; i--)

if (num%i!=0)

return 0;

return 1 ;

}

**int main()**

{

int x, found=0 ;

x=20 ;

do

{

found = checkX(x) ;

if (found==1)

printf("%d",x);

else

x+=20 ;

}

while (found==0) ;

}

**שאלה 7**

By listing the first six prime numbers: 2, 3, 5, 7, 11, and 13, we can see that the 6th prime is 13.

What is the 10 001st prime number?

על ידי רישום של ששת המספרים הראשוניים הראשונים: 2, 3, 5, 7, 11 ו- 13, אנו יכולים לראות שראש הממשלה השישי הוא 13.

מה המספר הראשוני ה-10,001 ?

**פתרון :**

* נתחיל מהמספר הראשוני הראשון, כלומר : 2. נריץ לולאה, שמתחילה ב-2 ומקדמת ב-1, בכל פעם. בודקת האם המספר ראשוני, ואם כן, סופרת אותו לתוך מונה ראשונים. מונה הראשונים הוא תנאי לעצירת הלולאה. כלומר, הלולאה תעצור ברגע שהמוה הגיע ל-10,001.
* בדיקת המספר האם הוא ראשוני נעשית באמצעות פונקציית עזר.

#include <stdio.h>

**int isPrimary(int num)**

{

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

{

if (num % i == 0)

{

return 0;

}

}

return 1;

}

**int main()**

{

int x=1,cnt=0;

while (cnt<10001)

{

x++;

if(isPrimary(x)==1)

cnt++;

}

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

}