אפקה המכללה האקדמית להנדסה בתל־אביב AFEKA TEL-AVIV ACADEMIC COLLEGE OF ENGINEERING

הנדסת תוכנה

Software Engineering

תרגיל 1 להגשה במבני נתונים (קורס מס' 10117)

מרצים : **ד"ר ראובן חוטובלי וד"ר דוד שטטר**

תאריך הגשה : 28.11.2020 עד השעה 23:00 . **העבודה - בזוגות**.

כל פונקציה/תכנית תכתב בשפת C .

שאלה מס' 1

נתונים שני מערכים של תווים, לא בהכרח זהים באורכם list1 ו-list2 .

כתבו פונקציה **רקורסיבית** אשר מקבלת 4 פרמטרים : 2 מערכים ואורכם.

void func (char list1[ ],char list2[ ],char list3[ ], int m, int n)

m – מציין את אורכו של list1 .

n – מציין את אורכו של list2 .

הפונקציה יוצרת משני מערכים הנתונים **מערך חדש** list3 באופן הבא :

האיבר הראשון במערך החדש יהיה האיבר הראשון במערך list1.

האיבר השני במערך החדש יהיה האיבר הראשון במערך list2.

האיבר השלישי במערך החדש יהיה האיבר השני במערך list1.

האיבר ה הרביעי במערך החדש יהיה האיבר השני במערך list2 . וכך הלאה....

במידה ואחד משני המערכים קצר יותר באורכו וסריקתו מסתיימת קודם, הפונקציה תשרשר את זנב המערך הארוך שנותר, לסוף המערך החדש.

דוגמה :

בעבור 2 המערכים האלה :

list1	'h'	'l'	'o'	'h'	'w'	'a'	'e'	'y'	'u'

list2	'e'	'l'	' '	'o'	' '	'r'	' '	'o'	' '	'!'

הערה : התו ' ' מסמל תו רווח.

הפונקציה תיצור את המערך החדש list3 שלהלן :

list3	'h'	'e'	'l'	'l'	'o'	' '	'h'	'o'	'w'	' '	'a'	'r'	'e'	' '	'y'	'o'	'u'	' '	'!'

שאלה מס' 2

סעיף א'

נתונה קבוצה של n סימנים המשוכנים במערך A שגודלו n.

נגדיר: f(n,k) כמספר האפשרויות לבחירת k סימנים שונים מקבוצת המספרים הנתונה כך שלא יבחרו שני סימנים המשוכנים במערך A במיקומים עוקבים. מהי הנוסחה הרקורסיבית בעבור f(n,k)?

1. $f(n,k)=f(n-2,k-1)+f(n-1,k)$.
2. $f(n,k)=\sum_{k=0}^{n/2} f(n-1,k)$.
3. $f(n,k)=1+f(n-2,k-1)+2+f(n-1,k)$.
4. $f(n,k)=1+f(n-2,k-1)+f(n-1,k)$

סעיף ב' נסמן f(k, n) את מספר האפשרויות לפיזור n כדורים זהים לתוך k תאים שונים **כך שבכל תא יכיל לכל היותר כדור אחד**. נוסחה רקורסיבית בעבור f(n,k) היא:

.1

$$f(k,n)=f(k-1,n)+f(k,n-1)$$
$$f(k,0)=1 \quad f(1,1)=1 \quad and \quad \forall n>1 \quad f(1,n)=0$$

.2

$$f(k,n)=f(k-1,n)+f(k-1,n-1)$$
$$f(k,0)=1 \quad f(1,1)=1 \quad and \quad \forall n>1 \quad f(1,n)=0$$

.3

$$f(k,n)=1+f(k-1,n)+f(k-1,n-1)$$
$$f(k,0)=1 \quad f(1,1)=1 \quad and \quad \forall n>1 \quad f(1,n)=0$$

4. אף אחת מבין התשובות הנתונות אינה נכונה.

סעיף ג'

מזכירה רשלנית שמה n מכתבים ב- n מעטפות ממוענות, באופן מקרי בלי לקרוא את הכתובות הרשומות על המעטפות. נסמן ב- f(n) את מספר האפשרויות שאף מכתב לא יגיע לתעודתו (לאדם שמיועד לו המכתב). נוסחה רקורסיבית (נסיגה) בעבור f(n) הינה:

1. f(n)=f(n-2)+f(n-1)
2. f(n)=n*(f(n-2)+f(n-1))
3. f(n)=(n-1)*(f(n-2)+f(n-1))
4. אף אחת מבין התשובות הנתונות אינה נכונה.

<u>סעיף ד'</u>

לציפי יש n שקלים, ובכל יום היא יכולה לקנות מסטיק ב-1 שקל או שוקולד ב-2 שקלים, או סוכריה ב-2 שקלים.

נגדיר ב- f(n) את מספר הדרכים שבהן יכולה ציפי לבזבז את כל כספה.

מהי הנוסחה הרקורסיבית בעבור f(n) ?

1. $f(n) = f(n-1) + 1 + 2f(n-2) + 4$
2. $f(n) = \sum_{i=1}^{\frac{n}{2}} f(n-i)$
3. $f(n) = 1 + f(n-1) + f(n-2)$
4. $f(n) = f(n-1) + 2f(n-2)$

<u>שאלה מס' 3</u>

לפניכם פונקציה **<u>רקורסיבית</u>** שכותרתה int what(int n1, int n2) המקבלת כפרמטרים שני מספרים שלמים והפונקציה תחזיר את הערך 1 אם הספרה השמאלית בשני המספרים האלה שווה, אחרת הפונקציה תחזיר את הערך 0.

לדוגמה:

- הקריאה הבאה What(286,25) תחזיר את הערך 1 .
- הקריאה הבאה What(286,65) תחזיר את הערך 0 .

```
int what(int n1, int n2)
{
        if (n1<10 && n2<10)
                return _____(1)____;
        if (n1<10)
                return _____(2)____;
        if (n2<10)
                return ______(3)____;
        _______(4)______;
}
```

בקוד הפונקציה חסרים **<u>ארבעה</u>** ביטויים, המסומנים במספרים בין סוגריים עגולים.
רשום על גבי גיליון התשובות את מספרי הביטויים החסרים (1) – (4), בסדר עולה, וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

<u>שאלה מס' 4</u>

כתבו פונקציה **<u>רקורסיבית</u>** אשר מקבלת כפרמטר num שבו מספר שלם בבסיס 10 ומחזירה מספר שלם שהינו ייצוג בינארי של num .

<u>שאלה מס' 5</u>

לפניכם תכנית בשפת C הכוללת פונקציה **<u>רקורסיבית</u>** אשר מקבלת כפרמטר num שבו מספר שלם בבסיס 10 **<u>ומחזירה מחרוזת</u>** המייצגת מספר שלם שהינו ייצוג הקסאדצימלי של num .

לצורך פתרון השאלה בתכנית זו אנו משתמשים בפונקציות, שאותן ראינו גם בשיעור לצורך הקצאות דינמיות.

להלן סיכום של הפעולות - הקצאה דינמית של זיכרון בשפת C .

פונקציה	תפקיד הפונקציה
sizeof (משתנה)	מחזירה גודל של משתנה או של טיפוס **<u>בבתים</u>**. לדוגמה: int i; result = sizeof (i); //if result == 4 → (int = 4 bytes)

malloc(size)	פונקציה זו נועדה להקצאת זיכרון דינמית: מספקים לפונקציה את גודל הזיכרון הנדרש בבתים (size) והיא מקצה זיכרון ומחזירה מצביע אליו.
	לדוגמה:
	int *i_ptr;
	i_ptr = (int *) malloc (sizeof (int));
	מקצים מקום בעבור int אחד ולמקום הזה מצביע i_ptr .
	**
	int* p=(int *)malloc(10*sizeof(int));
	מקצים מקום בעבור 10 int–יים ולמקום הזה מצביע p.

free(p)	שחרור השטח שעליו מצביע p כאשר השטח הוקצה באופן דינמי.

realloc(**…)**	מבצעת את הפעולות הבאות:
	1. הקצאת שטח דינמי חדש.
	2. העתקת תוכן של שטח ישן <u>שהוקצה דינמית</u> אל השטח החדש.
	3. שחרור השטח הישן.
	לדוגמה:
	int* p=(int*)malloc(2*sizeof(int));//integers הקצאת שטח עבור שני
	אפשר לומר ש-p הוא מערך של 2 int–ים.
	p[0]=7; p[1]=9;
	תמונת המצב הינה:

אינדקס	0	1
p - מערך	7	9

עתה אם נבצע את הפקודות שלהלן :

```
int size = 2 ;
p=realloc(p, ++size*sizeof(int));
```

נקבל :

המשתנה size מקבל ערך 3 ולכן מתבצעות הפעולות הבאות :

1. הקצאת שטח דינמי חדש של 3 int – ים. (**תא אחד יותר ממה שהיה קודם).**
2. העתקת תוכן של שטח ישן , שעליו מצביע הפרמטר הראשון של הפונקציה realloc שהוא – p , אל השטח החדש.

   תמונת המצב הינה :

אינדקס	0	1	2
	7	9	

3. שחרור השטח הישן , שעליו מצביע הפרמטר הראשון של הפונקציה realloc שהוא – p .
4. על השטח החדש יצביע המשתנה שמופיע בצד השמאלי (בצד המקבל) של פקודת ההשמה. בדוגמה זו המשתנה הוא p .

   לכן תמונת המצב הינה :

אינדקס	0	1	2
p	7	9	

עתה ב- p[2] אפשר לשים ערך כלשהו כרצוננו.

להלן התכנית הכוללת את הפונקציה ה**<u>רקורסיבית</u>**

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
char *s;
```

```
char *makenumintohexadecimal(int num)
{
	char dig2;
	int dig1;
	if (num == 0)
	{
		s = (char *)malloc(1);
		*s = '\0';
		return s;
	}
	dig1 = num % 16;
	if (dig1 < 9)
		dig2 = ______(1)______;
	else
		dig2 = ______(2)______;
	s = _____(3)_______;
	s = (char *)realloc(s, strlen(s) + 2);
	int len = _____(4)_____;
	_______(5)______;
	_______(6)______;
	return s;
}
void main()
	{
	int num, x;
	printf("insert a number:\n");
	scanf("%d", &num);
	printf("%s", makenumintohexadecimal(num));
	scanf("%d", &x);
	}
```

בקוד הפונקציה חסרים **שישה** ביטויים, המסומנים במספרים בין סוגריים עגולים.
רשום על גבי גיליון התשובות את מספרי הביטויים החסרים (1) – (6), בסדר עולה,
וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

שאלה מס' 6:

להלן הגדרה חדשה

חיפוש טרינארי הוא שיטת חיפוש אשר מנסה לאתר איבר X במערך ממוין.

בעבור המערך הממוין $A_0 \ldots\ldots A_{n-1}$ השיטה הזאת מחלקת את איברי המערך ל- 3 חלקים באופן הבא:

חלק 1 - יכיל את אברי המערך: $A_0 \ldots\ldots A_{\frac{n}{3}-1}$.

חלק 2 - יכיל את אברי המערך: $A_{\frac{n}{3}} \ldots\ldots A_{\frac{2n}{3}-1}$

חלק 3- יכיל את אברי המערך: $A_{\frac{2n}{3}} \ldots\ldots A_{n-1}$.

כדי לאתר את האיבר X בשיטת **חיפוש טרינארי** נבצע את הצעדים האלה:

צעד ראשון - אם האיבר X שווה לאיבר $A_{\frac{2n}{3}}$, אז האיבר X נמצא ונחזיר את מיקומו.

צעד שני - אם האיבר X גדול מאיבר $A_{\frac{2n}{3}}$, אזי תהליך חיפוש האיבר X בשיטה הטרניארית. נמשך **בחלק 3** של המערך.

צעד שלישי - אם האיבר X שווה לאיבר $A_{\frac{n}{3}}$, אזי האיבר X נמצא ונחזיר את מיקומו.

צעד רביעי - אם האיבר X גדול מאיבר $A_{\frac{n}{3}}$, אזי תהליך חיפוש האיבר X בשיטה הטרניארית. נמשך **בחלק 2** של המערך **-** אחרת תהליך חיפוש האיבר X בשיטה הטרניארית. נמשך **בחלק 1** של המערך.

כתבו בפונקציה רקורסיבית בעבור חיפוש טרינארי.

<u>שאלה מס' 7:</u>

במשחק מסוים משחקים שני שחקנים A ו-B. בתחילת המשחק לשחקן A יש m שקלים, ולשחקן B- n שקלים.

בכל שלב של משחק השחקן שמפסיד משלם שקל אחד למנצח. המשחק יסתיים כאשר אחד השחקנים יפסיד את כל כספו.

כך, למשל, אם בתחילת המשחק ל- A יש 3 שקלים, ול- B- 5 שקלים, הסדרה הבאה BBABB מתארת את התוצאות של המשחק (בשלב הראשון והשני המנצח הוא B, בשלב השלישי המנצח הוא A,ובשלב הרביעי והחמישי המנצח הוא B).

	תחילת המשחק	בתום שלב 1	בתום שלב 2	בתום שלב 3	בתום שלב 4	בתום שלב 5
הכסף שלרשות A	3	2	1	2	1	0
הכסף שלרשות B	5	6	7	6	7	8

ברור שבתום חמישה שלבים של המשחק, השחקן A הפסיד והשחקן B מנצח.

זהו סוג של משחק, שבו כל מה שמפסיד שחקן אחד עובר לשחקן השני.

תהי p הסתברות לכך, שהשחקן A ינצח בשלב כלשהו של המשחק.

כתבו נוסחה רקורסיבית (כולל תנאי עצירה) עבור ההסתברות, שהשחקן A יפסיד את כל כספו.

רמז לפתרון

נסמן ב- P_m את ההסתברות, שהשחקן A יפסיד במשחק כאשר הוא מתחיל את המשחק עם סכום כסף m שקלים-ואילו השחקן היריב B מתחיל עם n שקלים.

<u>שאלה מס' 8:</u>

לפניכם פונקציה **<u>רקורסיבית</u>** אשר מקבלת מספר שלם וחיובי – n (מותר להניח שהקלט חוקי) ומדפיסה מערך של 2^n מילים, כאשר כל מילה מיוצגת באמצעות מערך, בגודל n המורכב מאפסים ואחדים. סדר המילים הוא סדר המניה מאפס ל- 2^n-1 על בסיס 2.
הפונקציה הזו מבצעת קריאה לפונקציית עזר בה יש מספר פרמטרים נוספים.

למשל, עבור n=3 יש להדפיס:

0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

יש כאן שימוש בפונקציית עזר **<u>רקורסיבית</u>** שכותרתה bin(word, n, index) המקבלת:

- מערך word באורך n,
- אורכו של המערך, n, וכן
- מצביע index למיקום במערך שממנו ואילך אין תוכן ב-word,
  וממיקום זה הפונקציה משלימה **<u>באופן רקורסיבי</u>** את תוכן המילה הבינארית בכל האפשרויות, ומדפיסה אותן.

```
void print_array(int *a, int n)
{
  int i;
  for (i=0; i<n; i++) printf("%d ",a[i]);
  printf("\n");
}
void bin(int word[], int n, int index)
{
  if(______(1)_____)
 {
    print_array(word, n);
    return;
  }
  _____(2)_____;
  _____(3)______;
  ______(4)______;
  bin(word, n, index + 1);
}
```

הזימון לפונקציה יתבצע כך:

```
int word[3];
binary(word,3,0);
```

בקוד הפונקציה חסרים **ארבעה** ביטויים, המסומנים במספרים בין סוגריים עגולים.
רשום על גבי גיליון התשובות את מספרי הביטויים החסרים (1) – (4), בסדר עולה,
וכתוב ליד כל מספר את הביטוי החסר שהוא מייצג.

שאלה מס' 9:

אותה שאלה כמו בשאלה מס' 7 אך כל סימן במילה שייך לקבוצה $\{0,1, . . ., b\text{-}1\}$.

לדוגמה, עבור n=2 ,b=3 יש להדפיס:

```
0 0
0 1
0 2
1 0
1 1
1 2
2 0
2 1
2 2
```

המלצה: לכתוב פונקציית עזר **רקורסיבית** שכותרתה TR8(word, n, index)
המקבלת:

- מערך word באורך n,
- אורכו של המערך, n, וכן
- מצביע index למיקום במערך שממנו ואילך אין תוכן ב-word,

וממיקום זה הפונקציה משלימה **באופן רקורסיבי** את תוכן המילה בכל האפשרויות, ומדפיסה אותן.

שאלה מס' 10 – backtracking

כתבו תכנית רקורסיבית שתקלוט מספר שלם חיובי N (ניתן להניח שיתקבל כזה), ותדפיס את מספר כל האפשרויות של סדרת מספרים כך שסכום איברי הסדרה שווה ל- N ולאחר מכן את כל האפשרויות האלו.
כמו כן התכנית צריכה להדפיס את כל האפשרויות האלו בצורה ממוינת (ראה דוגמה של פלט)
דוגמה: עבור קלט 4 הפלט הוא:
מספר האפשרויות הוא: 8
והאפשרויות הן:

[1, 1, 1, 1]
[1, 1, 2]
[1, 2, 1]
[1, 3]
[2, 1, 1]
[2, 2]
[3, 1]
[4]

שאלה מס' 11 – backtracking - שאלת בונוס

בעיית שמונה המלכות

הבעיה: יש להציב שמונה מלכות על לוח שחמט,

כך שאף מלכה אינה מאיימת על רעותה.

כלומר:

אין שתי מלכות הניצבות באותה שורה.

אין שתי מלכות הניצבות באותה עמודה.

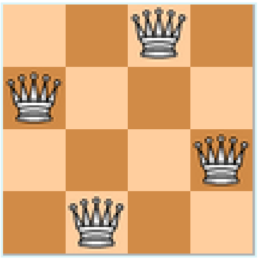
אין שתי מלכות הניצבות באותו אלכסון עולה.

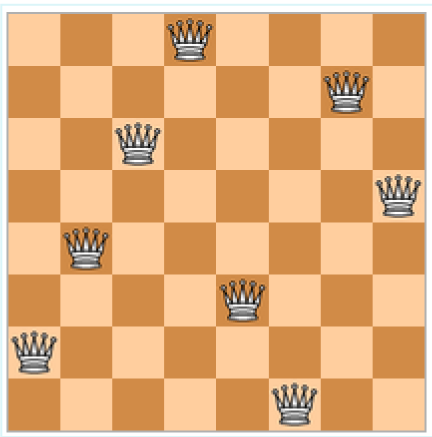
אין שתי מלכות הניצבות באותו אלכסון יורד.

נדון בהכללה של בעיה זו : N מלכות ולוח NXN.

דוגמאות :

- עבור שלוש מלכות ולוח בגודל 3×3 אין לבעיה פתרון.

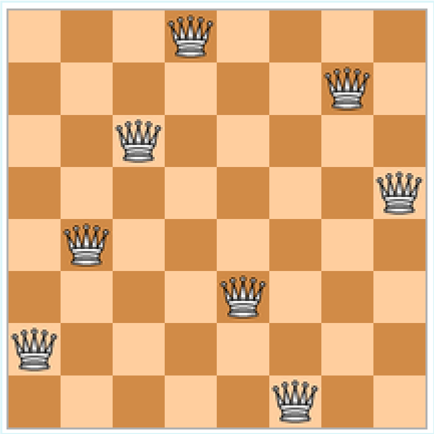
עבור ארבע מלכות ולוח בגודל 4×4 קיימים 2 פתרונות. לדוגמה :	

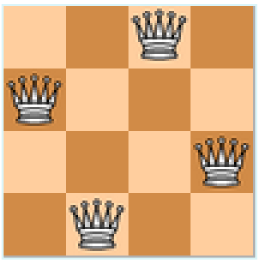
עבור שמונה מלכות ולוח בגודל 8×8, קיימים 92 פתרונות. לדוגמה :	

בכל עמודה (ובכל שורה) עומדת מלכה בודדת.

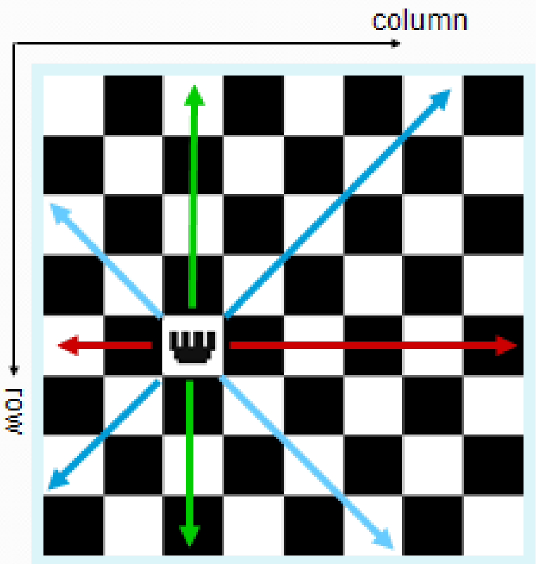
נייצג את לוח השחמט $N \times N$ באמצעות מערך

הערך הנמצא ב- **board[i]** הוא מספר השורה בה נמצאת המלכה בעמודה ה-**i**.

עבור שמונה מלכות ולוח בגודל 8×8, שלהלן נחזיק מערך board שלהלן: **{6, 4, 2, 0, 5, 7, 1, 3}** הניחו שהאינדקס מתחיל ב- 0 .	

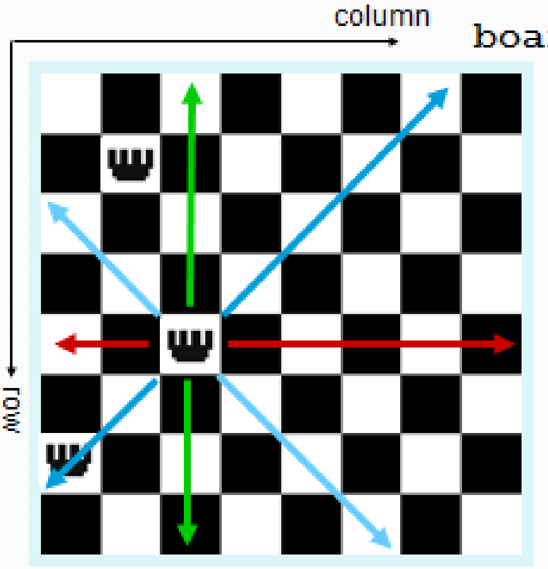
עבור שמונה מלכות ולוח בגודל 4×4, שלהלן נחזיק מערך board שלהלן: **{1, 3, 0, 2}** הניחו שהאינדקס מתחיל ב- 0 .	

- עליכם להחזיק דגלים (**used**) שיאפשרו בדיקת חוקיות של מיקום חדש.



- בשל דרך אחזקת הלוח ע"י **board**, כשאנו בוחנים את האיבר ה-**i**, ברור כי הוא בעמודה פנויה.
- עליכם לבדוק אם השורה ושני האלכסונים פנויים. לכן נחזיק שלושה מערכי **used** (אדום, תכלת, וכחול).
- מערכים אלו יאותחלו באפסים, ויעודכנו ל- 1 בהיתפס שורה\אלכסון\עולה\אלכסון יורד.

- נניח כי הצבתם מלכות בהצלחה בעמודות `0,1,2, … , column-1`.
- בהינתן הצבה `board[column]=row` יהיה עליכם לבדוק 3 בדיקות:
  - השורה – `?used1[row]==1`
  - אלכסון עולה – `?used2[7-column+row]==1`
  - אלכסון יורד – `?used3[row+column]==1`
- אם התנאים מתירים הצבה זו, נידרש לעדכן את מערכי הדגלים בהתאם.
- עם שחרור מקום יש לעדכן הדגלים באופן הפוך (ל-0).



כיצד מומלץ לבחור שורה עבור המלכה בעמודה הנוכחית **column**?

מומלץ לבחון כל שורה מ-**0** ועד **N-1**.

אם בשורה המוצעת המלכה אינה מאיימת על אף מלכה קודמת, נבחר שורה זו ונמשיך בהתקדמות.

לאחר מציאת כל הפתרונות האפשריים במיקום זה, ננסה שורה נוספת (לצורך מציאת פתרונות נוספים) .

אם המלכה בשורה המוצעת מאיימת על אחת המלכות הקודמות, נפסול אפשרות זו וננסה שורה אחרת.

אם לא נותרו שורות אפשריות נוספות, ניסוג על עץ האפשרויות, כלומר נחליף את מיקום המלכה הקודמת.

הגדירו את :

— **#define N 8**

— **#define DIAG (N*2 - 1)**

עליכם לכתוב פונקציה רקורסיבית שכותרתה :

```
void TR_Solutionsr (int board[n], int column,  bool used[3][DIAG]);
```

כאשר פונקציית ה- ()main תראה כך :

```
int main()
{
  int board[N];
  int used[3][DIAG]; // row, up-diag, down-diag
    //Initializing used arrays. Note that there are
  //only N rows so the remaining elements
  //in used[0] will be discarded.
  for(int i = 0; i < DIAG; i++)
    used[0][i] = used[1][i] = used[2][i] = FALSE;
  printf("Solving the %d Queens Problem\n\n", N);
  TR_Solutions(board, 0, used);
  printf("\nDone!\n");
  return 0;
}
```

בהצלחה