

מבוא לתכנות

הרצאה 8 – מערכים דו ממדיים - מטריצה
סמסטר 1

מערך

- איברים במערך מסודרים ומאונדקסים כמו במחרוזות.

– האינדקס של האיבר הראשון הינו 0

– האינדקס של האיבר האחרון במערך עם גודל n הינו $(n-1)$

- כדי לעדכן איבר באינדקס ספציפי, נשתמש ב:

```
names[1] = "Bob"; //write to the second element  
names[0] = "Alice"; //write to the first element  
names[3] = "Dora"; //write to the last element
```

- או, שנגדיר ונעדכן את כל האיברים בבת אחת כך:

```
int[] names = new string[]{"Alice", "Bob", "Charlie", "Dora" };
```

סוגי נתונים

- יש שני סוגי של נתונים:

הפניה	ערך
string	int
array	float\double
	boolean
	char

- סוגי "ערך" מכילים ערך יחיד ופשוט.
- סוגי "הפניה" מחזיקים נתונים מורכבים:
 - מחרוזת: רצף תווים
 - מערך: רצף של ערכים

סוגי נתונים

Variable	Value
x	17
numbers	

הפניה

Value
1
2
3

- כאשר שמים ערך במשתנה, המשתנה פשוט מחזיק את הערך:

```
x = 17
```

- אבל כאשר שמים הפניה למשתנה, המשתנה מכיל הפניה לאובייקט:

```
int[] numbers = new int[] {1,2,3};
```

הפניות מרובות

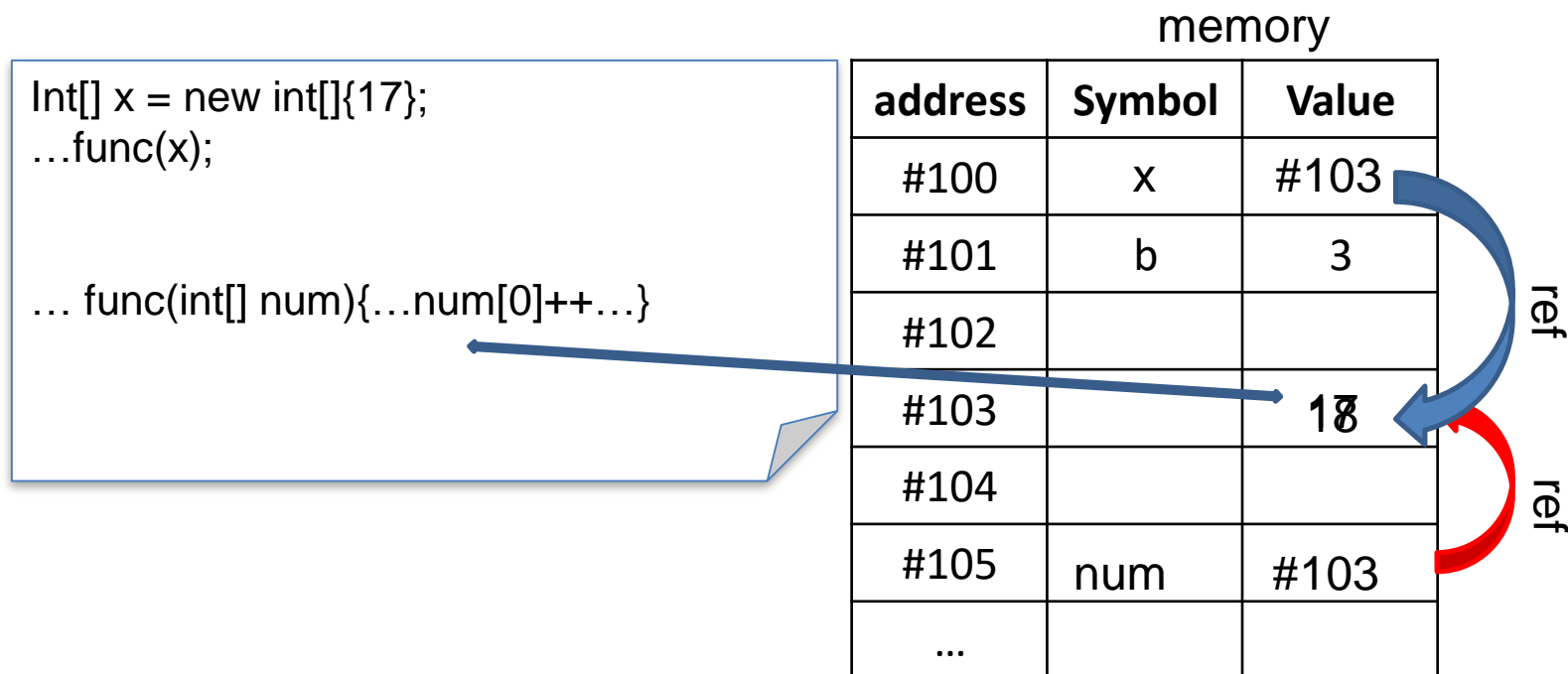
- שימו לב לקוד הבא:

```
int[] a = new int[]{1, 2, 3, 4, 5};  
int[] b = a;  
b[2] = 7;
```

- מהם הערכים ב-a?
- מהם הערכים ב-b?
- זה נראה כאילו a ו-b הם מערכים נפרדים, אך הם לא.
- למעשה, a ו-b הם הפניות לאותו המערך!
- אם נשנה משהו ב-b, נשנה גם את a. מדוע? בגלל ש-a ו-b מפנים לאותו מערך.
- כמו כן גם הפוך, שינויים ב-b ישנו את a.
- הפניות מרובות היא טכניקת תכנות חשובה שנשתמש בה בהמשך הקורס

סוג הפניה

- כשמשתמשים בסוג הפניה (למעט מחרוזת) כארגומנט, אנו שולחים את ההפניה ולכן הערכים המקוריים של המערך ישתנו בתוך הפונקציה
- מאחורי הקלעים...



מערך דו-ממדי

- מערך יכול להכיל כל סוג של ערך
- זה כולל, כמובן, גם מערכים!
- דוגמה:

```
int[,] a = new int[2, 3];  
int[,] b = new int[2, 3] { { 1, 2, 3 }, { 4, 5, 6 } };  
int[,] c = new int[,] { { 10, 20, 30 }, { 40, 50, 60 } };
```

– מה יחזיר *b.Length* ?

– מה יחזיר *c[1,1]* ?

מערכים דו-ממדיים

```
int[,] a = new int[,] { { 1, 2, 3, 4 }, { 5, 6, 7, 8 }, { 9, 10, 11, 12 } };
```

- נוכל לפנות ל $a[0, 1]$ ונקבל 2
- נוכל לראות ברשימה דו-ממדית זו כטבלה, הקרויה בשם "מטריצה"

1	2	3	4
5	6	7	8
9	10	11	12

- נוכל לגשת לתא (שורה, עמודה) של המטריצה באמצעות:
 $a[\text{row}, \text{column}]$

– row הוא אינדקס השורה, column הוא אינדקס העמודה.
(כרגיל, האינדקסים מתחילים באפס)

הפניות והשמות

```
int[,] a = new int[,] { { 1, 2, 3 }, { 4, 5, 6 } };  
int[,] b = a;  
b[1, 2] = 7;
```

```
b = new int[1,1];  
Console.WriteLine(a[1,1]);  
a = new int[2, 2];
```

1,2,3,
4,5,7,

5

0,0,
0,0,

- זיכרו: רשימה היא הפניה!

Length and GetLength()

- Length - מחזיר את כמות האיברים במערך
- GetLength(int dim) - מחזיר את הגודל של המימד המבוקש

```
int[,] a = new int[2,3];  
Console.WriteLine(a.Length);  
Console.WriteLine(a.GetLength(0));  
Console.WriteLine(a.GetLength(1));
```



6
2
3

תרגיל 1.1

- צרו והדפיסו את לוח הכפל הבא:

```
1  2  3  4  5  6  7  8  9 10
2  4  6  8 10 12 14 16 18 20
3  6  9 12 15 18 21 24 27 30
4  8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
Press any key to continue . . .
```

- התחילו בכתיבת פונקציה בשם

```
int[,] CreateMultiplicationBoard()
```

אשר מחזירה מערך דו-ממדי.

תרגיל 1.2

- כעת, כיתבו פונקציה בשם

`PrintBoard(int[,] mat)`

שמדפיסה את המערך הדו-ממדי אשר מתקבל כפרמטר.

תרגיל 2

- בהינתן ערכי המטבע הבאים, נחשב את ממוצע הערכים בעמודה החסרה. ברשותנו מערך דו-ממדי עבור הערכים ומערך חד-ממדי עבור שמות המטבעות.

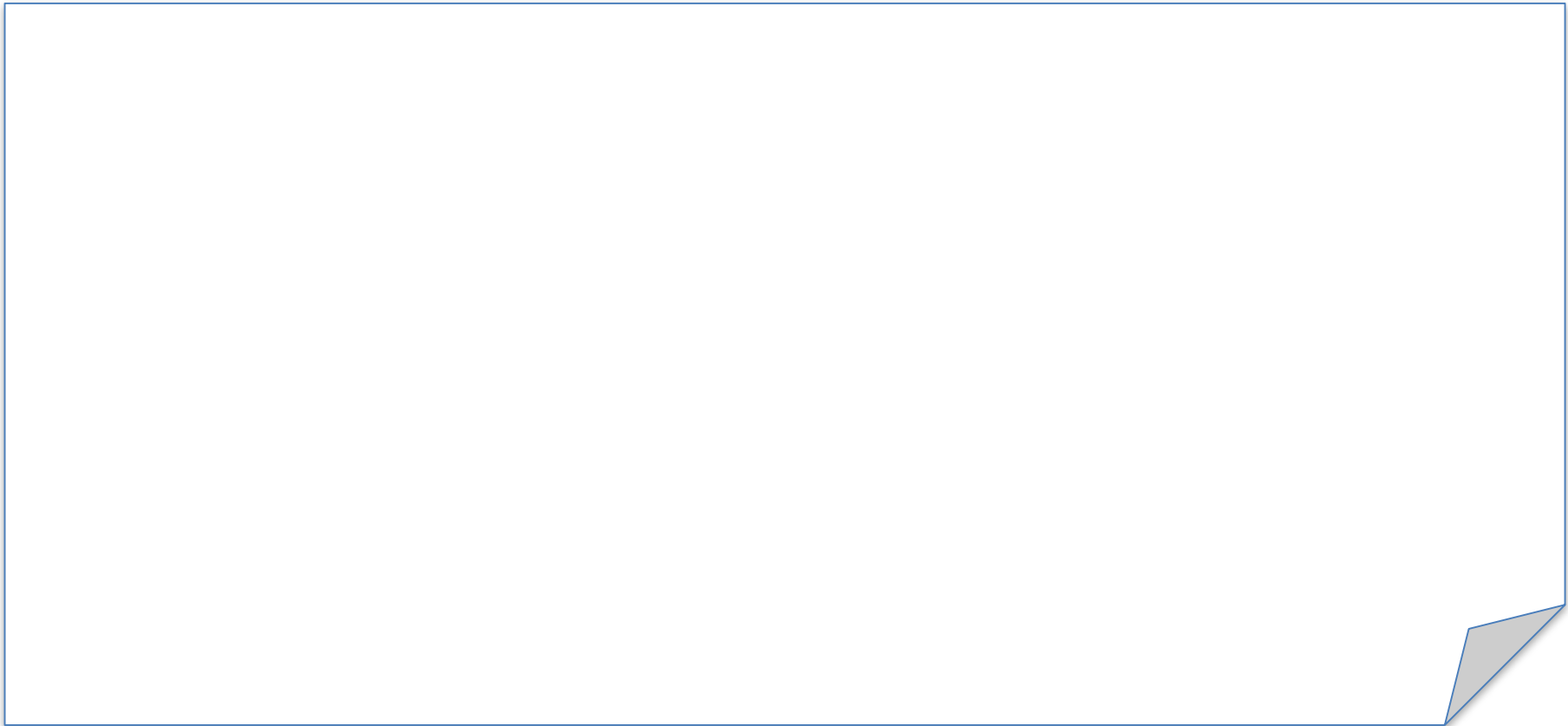
Currency	Jan	Feb	March	avg
Dollar	3.544	3.691	3.623	?
Euro	4.73	5.0823	5.0059	?
Yen	4.3551	4.5272	4.4113	?
GBP	5.493	5.9473	5.8959	?

- הטבלאות להזנה הינן:

```
double[, ] cur = new double[, ] { {3.544, 3.691, 3.623, 0},  
                                     {4.73, 5.0823, 5.0059, 0},  
                                     {4.3551, 4.5272, 4.4113, 0},  
                                     {5.493, 5.9473, 5.8959, 0} };  
string[] currNames = new string[] { "Dollar", "Euro", "Yen",  
                                     "GBP" };
```

תרגיל 2.1

- ראשית כיתבו פונקציה בשם `string`
`TablString(double[,] table, string[]
names)` שמחזירה מחרוזת להדפסה של המערכים
שהתקבלו כפרמטרים.



תרגיל 2.2

- כעת כיתבו פונקציה בשם
`AverageRows (double[,] table)`
אשר מקבלת את טבלת ערכי המטבע ומעדכנת את
העמודה האחרונה להיות הממוצע של כל שורה.

תרגיל 2.3

- אתם צריכים לקנות בכל חודש 1 דולר, אירו, ין ופאונד.
ממשו את הפונקציה `double[]`
`SumColumns(double[,] table)` אשר מחשבת
את הסכום של כל עמודה ומחזירה רשימה של כל
הסכומים.

Rank And Clone()

• Rank - יחזיר את מספר הממדים

```
int[,] arr = new int[,] {{1,2,3}, {4,5,6} };  
Console.WriteLine(arr.Rank);
```

2

• Clone() - יחזיר עותק נוסף של המטריצה

```
int[,] arr = new int[,] {{1,2,3}, {4,5,6} };  
int[,] arr2 = (int[,])arr.Clone();  
arr[1, 2] = 7;  
Console.WriteLine(arr2[1,2]);
```

6

- נוכל להיעזר בהערות מסוג `///` כדי לקבל פורמט מקצועי ונחמד עבור ההערות מעל הפונקציה וקטעי קוד אחרים

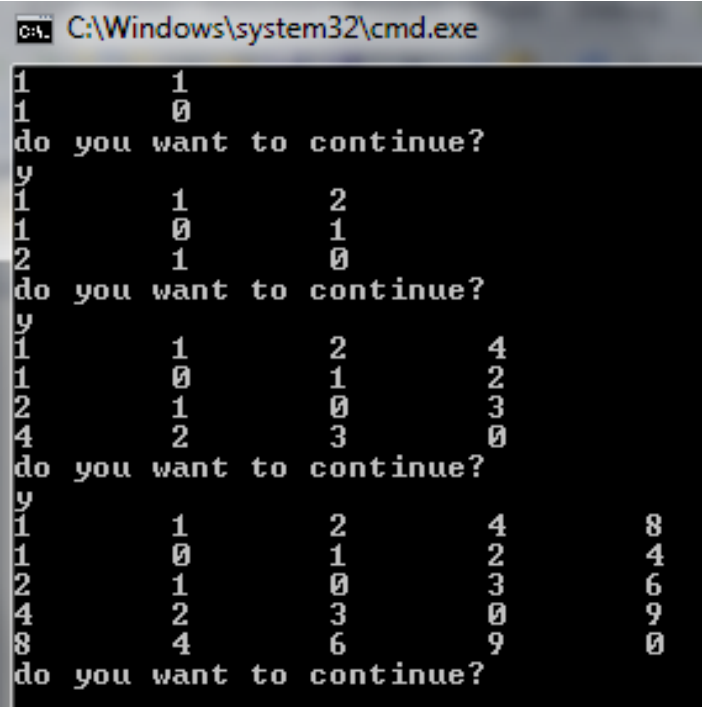
תרגיל 3

- נתחיל במטריצה בגודל 1×1 עם הערך 1 ונגדיל אותה בצעדים קבועים. כל הגדלה תוסיף שורה ועמודה. הערך בשורה/עמודה החדשים יהיה הסכום של כל הערכים בשורות/עמודות שהיו עד כה.

• נעשה זאת באמצעות 4 הפונקציות הללו:

- Main
- void PrintBoard(int[,] mat)
- int[,] PumpMatrix(int[,] matrix)
- void CopyMatrixValues(int[,] sourceMatrix, int[,] destinationMatrix)

- הניחו שאין לנו את פונקציות השכפול/שיבוט.



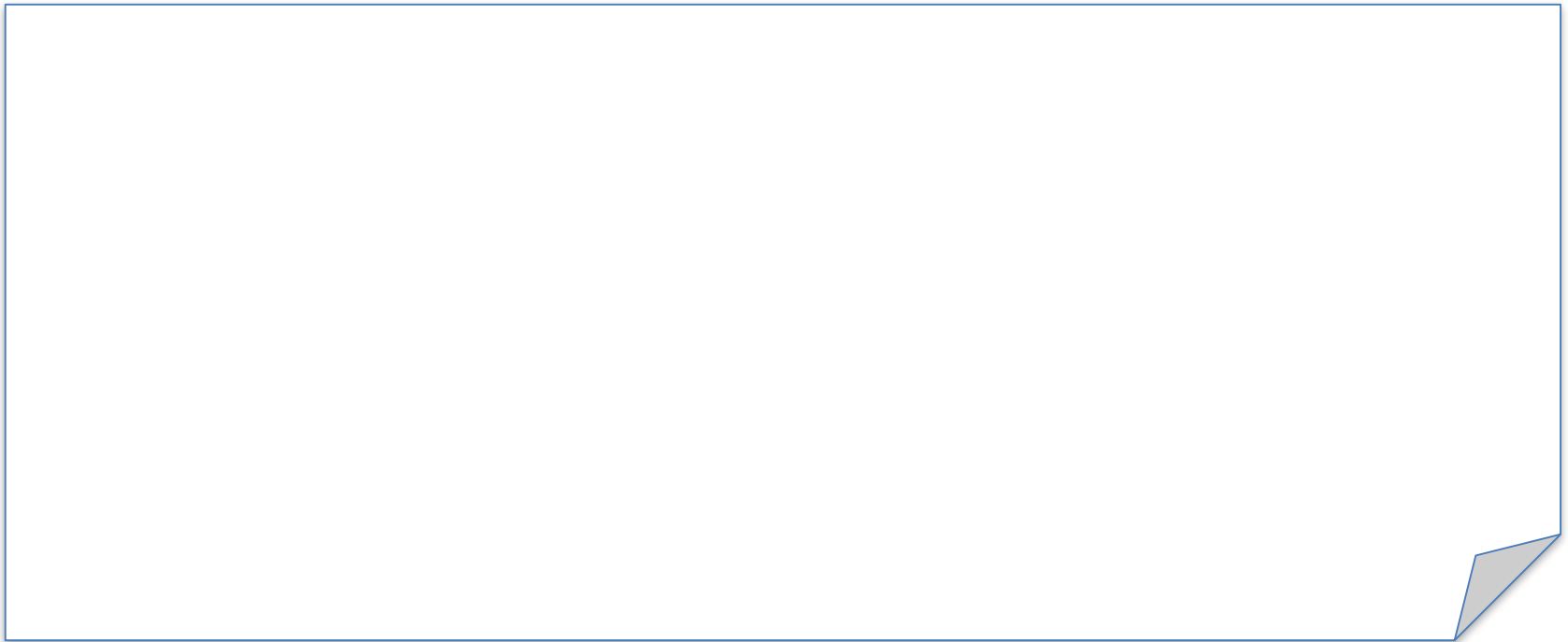
```
C:\Windows\system32\cmd.exe
1      1
1      0
do you want to continue?
y
1      1      2
1      0      1
2      1      0
do you want to continue?
y
1      1      2      4
1      0      1      2
2      1      0      3
4      2      3      0
do you want to continue?
y
1      1      2      4      8
1      0      1      2      4
2      1      0      3      6
4      2      3      0      9
8      4      6      9      0
do you want to continue?
```

תרגיל 3

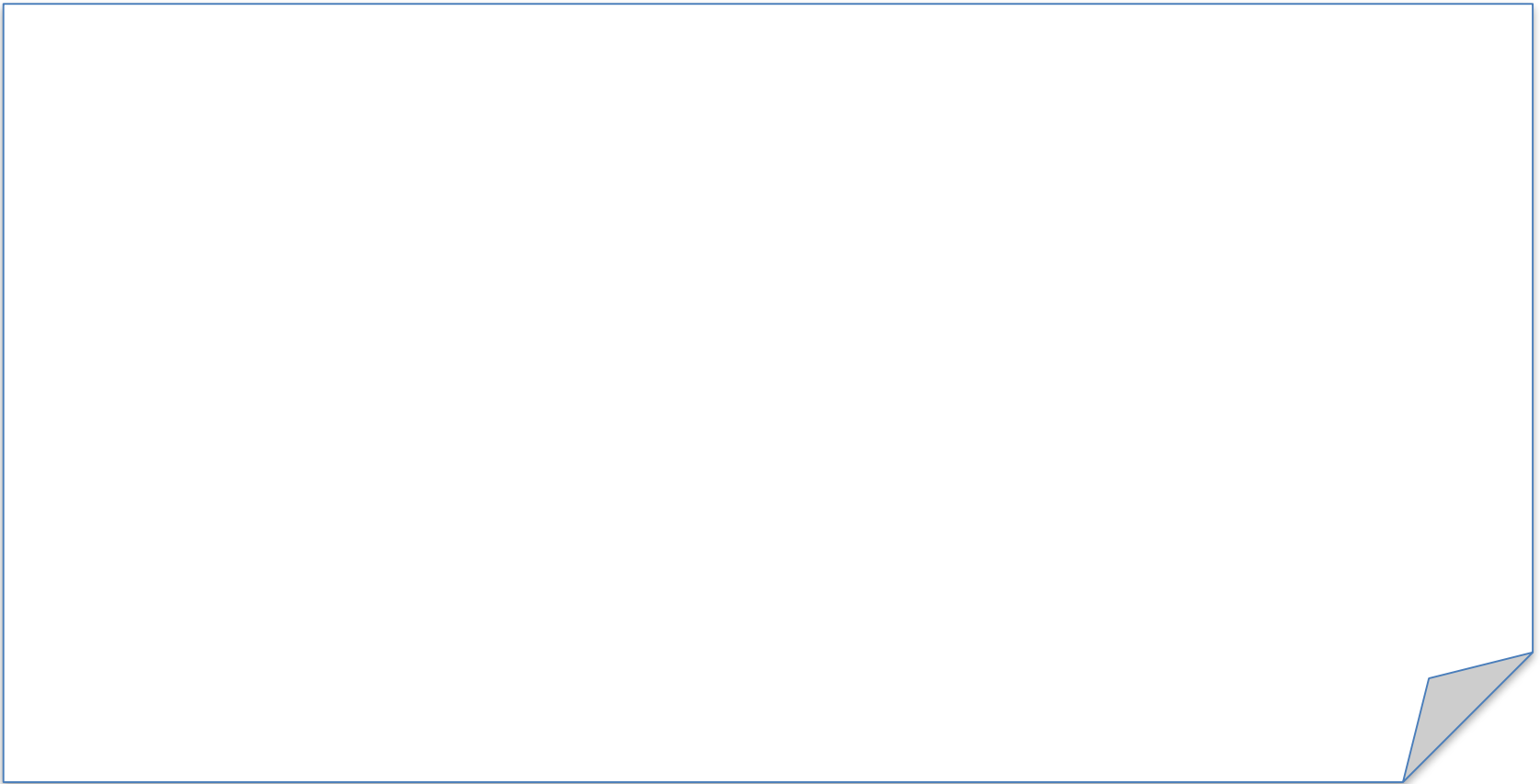
- ראשית ניצור את

```
void PrintBoard(int[,] mat)
```

זה ידפיס את המטריצה.



- כעת נעבור לפונקציית Main



תרגיל 3

• כעת נעבור ל

```
void PrintBoard(int[,] mat)
```

תרגיל 3

- וכעת ניצור את

```
int[,] PumpMatrix(int[,] matrix)
```

תרגיל 3

- נותר רק להשלים את

```
void CopyMatrixValues(int[,]  
sourceMatrix, int[,] destinationMatrix)
```