

סמסטר א' 20??
מועד ?

מרצה : ניר חן

1. _____

2. _____

3. _____

4. _____

סה"כ : _____



הוראות כלליות:

1. משך הבחינה **שלוש שעות**.
2. כל חומר עזר אסור לשימוש. למעט מחשבון.
3. יש לענות על כל השאלות על גבי טופס המבחן.
4. אם אינכם מסוגלים לתת תשובה מלאה לשאלה כלשהי, תנו תשובה חלקית. תשובה נכונה באופן חלקי תקבל ניקוד באופן חלקי.
5. אם אינכם זוכרים את השם או הסינטקס המדויק של פקודה כלשהיא אפשר להמציא שם וסינטקס סבירים וזאת בתנאי שתוסיפו תיאור מדויק של הפקודה בנוסף לפתרון.
6. התשובות חייבות להיות קצרות ולעניין. כתב היד חייב להיות ברור וקריא. תשובות לא קריאות תקבלנה ציון 0. כמו כן יש ליישר את הקוד בהתאם למוסכמות התכנות שלמדתם בקורס.
7. אם התבקשתם לכתוב תכנית או פונקציה שאמורה לפעול על קלט מסויים, אזי התוכנית או הפונקציה לא אמורה לבדוק אם הקלט תקין, אלא אם כן נאמר כך בשאלה במפורש.
8. הקוד שתכתבו ישפט, בין היתר, לפי האורך והאלגנטיות שלו. פונקציות או תוכניות ארוכות או מסורבלות ללא צורך יקבלו פחות נקודות, אפילו אם הן ממלאות כהלכה את המשימה שהוגדרה בשאלה.
9. יש להשתמש בשמות משתנים בעלי משמעות!

בהצלחה!

ישנם מספר דפי טיוטה בסוף מחברת הבחינה

ענה/י על השאלות הבאות:
שאלה 1 (25 נקודות)

עליך לממש את הפונקציה

```
int RotateDigits(int digits)
```

הפונקציה מקבלת מספר כלשהוא ומחזירה מספר שסדר ספרותיו הפוך.

לדוגמה:

```
int number = 36812;  
int newNumber;  
newNumber = RotateDigits (number);  
Console.WriteLine(newNumber);
```

הפלט:

21863

אין להשתמש במערכים, מחרוזות והפונקציות שלהם!!!
עבור תשובה ריקה לגמרי יתקבלו 7 נקודות. עבור תשובה שגויה לגמרי לא יתקבלו נקודות כלל!!!

שאלה 2 (20 נקודות)

השלם את החסר אך ורק איפה שיש קו תחתון:
 הפונקציה מקבלת מערך דו מימדי בגודל 4 על 4, מלא בערכים מ 0 עד 5 (כולל 0 ו 5).
 הפונקציה בודקת האם מתקיימת סימטריות בין 4 חלקי המטריצה.
 דוגמה למטריצה סימטרית

1	2	2	1
3	4	4	3
3	4	4	3
1	2	2	1

```
bool IsSymmetrical(int[, ] numbers)
{
    //init.
    bool result = ____;

    //checking symmetric property.
    for (int i = 0; i < numbers.GetLength(0) / ____; i++)
    {
        for (int j = 0; j < numbers.GetLength(1) / ____; j++)
        {
            if (numbers[i, j] ____ numbers[numbers.GetLength(0) - 1 - i, j])
            {
                result = ____;
            }

            if (numbers[i, j] != numbers[____, ____])
            {
                result = ____;
            }

            if (numbers[i, j] ____ numbers[numbers.GetLength(0) - 1 - i, ____])
            {
                result = ____;
            }
        }
    }

    return ____;
}
```

הפלט עבור המטריצה הסמטרית:

true

שאלה 3 (30 נקודות)

* (בבחינה תנתן שאלה בסגנון הזה אבל היא תהיה מורחבת יותר ותכיל יותר מתודות למימוש כמו שעשינו ב (Vector, Stack, Queue, Linked list, Peron, Car...))

עליך לממש את המחלקה (class) הבאה:

עבור כל פונקציה שתשובתה ריקה לגמרי יתקבלו 2 נקודות. עבור כל פונקציה שתשובתה שגויה לגמרי לא יתקבלו נקודות כלל!!!

```
//write a class that generates primes
public class PrimeGenerator
{
    //members
    int Position;    // The current position of the prime generation

    //constructor
    // Constructs a new prime generator, the generator is initialized to position 1
    public PrimeGenerator()
    {

    }

    // Sets a new position, the generator will returns the primes following this new position
    value
    public void SetPosition(int NewPosition)
    {

    }

    //returns the current position
    public int GetPosition()
    {

    }

    // Returns the next prime of the generator. The next prime is the first number which is
    prime after the //position value
    public int NextPrime()
    {
```

```
}
}
```

ניתן להשתמש במחלקה באופן הבא:

```
PrimeGenerator generator;
generator = new PrimeGenerator();
int prime = generator.NextPrime();
Console.WriteLine("A prime : " + prime.ToString());
generator.SetPosition(10);
prime = generator.NextPrime();
Console.WriteLine("A prime : " + prime.ToString());
prime = generator.NextPrime();
Console.WriteLine("A prime : " + prime.ToString());
prime = generator.NextPrime();
Console.WriteLine("A prime : " + prime.ToString());
```

התוצאה תהיה:

```
"A prime : 2"
"A prime : 11"
"A prime : 13"
"A prime : 17"
```

עוד דוגמה לשאלה 3 (30 נקודות)

```
/// <summary>
/// the class person will declare three members the person's name, his energy and
/// the path he had already run in his life.
/// the person can eat some food portion and so gain energy.
/// the person can run and burn the energy. Every distance unit run will burn one portion
/// of energy.
/// a person with no energy (energy=0) is considered dead!
/// </summary>
public class Person
{
    private int path;    //how much distance units the person run;
    private int energy;  //how many energy the person has = portions of food that the
person eat.
    private string name;    //the person's name

    /// <summary>
    /// this constructor will assign the name to be the empty string,
    /// the path to zero and the energy to 1
```

```
/// </summary>
public Person()
{
```

```
}
```

```
/// <summary>
/// every food portion will increase the persons energy by one.
/// two portions will increase by two, and so on...
/// </summary>
/// <param name="portions"></param>
public void Eat(int portions)
{
```

```
}
```

```
/// <summary>
/// will return the information whether the person has energy that is equal to zero
/// that is considered dead.
/// </summary>
/// <returns>dead returns true, alive returns false </returns>
private bool IsDead()
{
```

```
}
```

```
/// <summary>
/// will change the name of the person
/// </summary>
/// <param name="n1">the new name</param>
/// <returns>the string "the new name is: X", where X in the new name. </returns>
public string ChangeName(string n1)
{
```

```
}
```

```

    /// <summary>
    /// if the person is alive will run one distance unit and burn one food portion.
    /// if the person is dead will print to the console "can not run, person is dead!"
    ///
    /// </summary>
    public void Run()
    {

    }

    /// <summary>
    /// will print to the console the 3 members, each in a different line and
    /// the name of the parameter
    /// </summary>
    public void PrintPerson()
    {

    }
}

```

ניתן להשתמש במחלקה באופן הבא

```

Person p1 = new Person();
Console.WriteLine( p1.ChangeName("ArikAndBentz") );
Console.WriteLine();
p1.PrintPerson();
p1.Run();
p1.Run();
p1.Eat(3);
p1.PrintPerson();
p1.Run();
p1.PrintPerson();
התוצאה תהייה
the new name is:ArikAndBentz

```

name:ArikAndBentz

energy:1

path:0

can not run, person is dead!

name:ArikAndBentz
energy:3
path:1
name:ArikAndBentz
energy:2
path:2

שאלה 4 (25 נקודות)

לכל אחת מהטענות הבאות, השיבו נכון/לא נכון ונמקו בקצרה:
(תשובה ללא נימוק לא תקבל נקודות כלל)

א. (5 נקודות) התבוננו בקטע הקוד הבא:

```
int[] arr = new int[] {2,3,5,1,4 };  
Console.WriteLine("START");  
foreach (int num in arr)  
{  
    num++;  
    Console.Write(num + ", " );  
}  
Console.WriteLine("END");
```

האם הקוד ירוץ ללא שגיאות? עבור תשובה "נכון", נא לפרט מהי השגיאה בקוד.
נכון / לא נכון? נמקו.

ב. (5 נקודות) קומפיילר הוא תוכנית המתרגמת קוד משפה עילית (high level language) לשפת מכונה (low level language / machine language)

נכון / לא נכון? נמקו.



(סוף המבחן)



ג. ...
ד. ...
ה. ...

int grade		הגדרת משתנה
<pre>if (grade > 60) { Console.WriteLine("Pass") } else { Console.WriteLine("Fail") }</pre>		משפט if
<pre>switch (a) { case 0: Console.WriteLine("zero"); break; case 1: Console.WriteLine("one"); break; default: Console.WriteLine("default"); break; }</pre>		משפט switch
s.EndsWith("ing")	→ true	string s = "programming"
s.StartsWith("pr")	→ true	
s.IndexOf("g", 6)	→ 10	
s.Insert(0, "the ")	→ "the programming"	
s.Remove(1, 2)	→ "pgraming"	
s.Replace("ing", "")	→ "program"	
s.Substring(4, 3)	→ "ram"	
i = Convert.ToInt32(d)	→ i = 7	int i = 5, double d = 7.7, string s = "2"

<code>i = Int32.Parse(s)</code>	$\rightarrow i = 2$	
<code>s = i.ToString()</code>	$\rightarrow s = "5"$	
<code>for (int i = 1; i < 10; i++) { Console.WriteLine(i); }</code>		לולאת for
<code>while(a < 4) { a = a + 1; }</code>		לולאת while
<code>int[,] integerArray = new int[10,5]; integerArray.Length; integerArray.GetLength(0); integerArray.GetLength(1);</code>		הגדרת מערך
<code>-\n: enter a new line -\t: insert a tab -\b: one char back -\\: print the char \ -\" : print the char "</code>		Special characters
<code>int.Parse (Console.ReadLine()) double.Parse (Console.ReadLine())</code>		Converting inputs
<code>(int) (double)</code>		casting
<code>.CopyTo(array, index) .Copy(src, dest, length) .Clone() – we need to cast the return type.</code>	<code>*Src = the source array *Dest = the destination array *length = amount of elements to copy</code>	Copy arrays
<code>Array.Sort()</code>	It sorts the elements in the array according to their type and order	Sorting an array
<code>Clear(array,index,length)</code>	Index – where to start Length – how many elements to clear	Array clear
<code>Foreach (type varName in ArrName) { Use varName here as the current value }</code>		foreach
<code>[array name].Rank</code>	will return the number of dimensions –in our	rank
<code>(casting)[array name].Clone()</code>	will make and return another copy of the matrix	clone
LIFO – last in, first out	Push – add an element Pop – remove an element	stack

FIFO – first in last out	Enqueue – add an element Dequeue - remove an element	Queue
--------------------------	---	-------