GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: אביב 2023 מועד ב'
סמל השאלון: 97105
שם התלמיד/ה: _____

מבנה נתונים ותכנות מונחה עצמים

הנדסת תוכנה · הנדסאים וטכנאים

הוראות

א. משך הבחינה: ארבע שעות וחצי.

ב. מבנה השאלון ומפתח ההערכה: **בשאלון זה שני מבחנים, עליכם לענות על מבחן אחד בלבד בהתאם למוסד הלימודים:** מבחן ב- Java (עמוד 2) מבחן ב- C# (עמוד 14) בכל מבחן 10 שאלות. **חלק א' – 48 נקודות** שאלות 1-4: יש לענות על **שלוש** שאלות בלבד. ערך כל שאלה 16 נקודות. **חלק ב' – 36 נקודות** שאלות 5-8: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה 18 נקודות. **חלק ג' – 16 נקודות** שאלות 9-10: יש לענות על **אחת** בלבד. ערך שאלה 16 נקודות. **בסך הכול: 100 נקודות.**

ג. חומר עזר מותר בשימוש: 1. מחשבון (אין להשתמש במחשב כף יד או במחשבון עם תקשורת חיצונית). 2. קלסר **אחד בלבד** עם **חומר ההרצאות**. אין להוציא דפים מהקלסר. **אין לצרף ספרים או חוברות עם פתרונות.**

ד. הוראות מיוחדות: 1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות הבחינה, ולוודא שהן מובנות. 2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי (גם בכך תלויה הערכת הבחינה). 3. יש להשאיר את העמוד הראשון במחברת הבחינה ריק. בסיום המבחן יש לרשום בעמוד זה את מספרי התשובות לבדיקה. התשובות ייבדקו לפי סדר כתיבתן בעמוד זה. לא ייבדקו תשובות עודפות. 4. יש לכתוב את התשובות במחברת הבחינה **בעט בלבד**, בכתב יד ברור. 5. יש להתחיל כל תשובה בעמוד חדש ולציין את מספר השאלה ואת הסעיף. אין צורך להעתיק את השאלה עצמה. 6. טיוטה יש לכתוב במחברת הבחינה בלבד. יש לרשום את המילה "טיוטה" בראש העמוד ולהעביר עליו קו כדי שלא ייבדק. 7. יש להציג פתרון מלא ומנומק, כולל חישובים לפי הצורך. הצגת תשובה סופית ללא שלבי הפתרון לא תזכה בניקוד. 8. יש להסביר בפירוט כל תוכנית שנכתבה, **תוכנית ללא הסבר מפורט לא תזכה בניקוד.** 9. אם לדעתכם חסר בשאלה נתון, יש לציין זאת ולהוסיף נתון מתאים שיאפשר לכם להמשיך בפתרון השאלה, נמקו את בחירתכם.

השאלות בשאלון זה מנוסחות בלשון רבים, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

השאלות

1. ציוני תלמידים שמורים בתור כך שציוניו של תלמיד מסויים מופרדים מציונים של תלמיד אחר עם הערך -1.

לדוגמה:

תור ציונים marks:

תלמיד ראשון השלים שלוש בחינות וקיבל את הציונים הבאים: 80, 90, 100.

תלמיד שני השלים שתי בחינות וקיבל את הציונים הבאים: 75, 96.

תלמיד שלישי השלים ארבע בחינות וקיבל את הציונים 100, 100, 97, 96.

תלמיד רביעי לא השלים אף בחינה ואין לו ציונים.

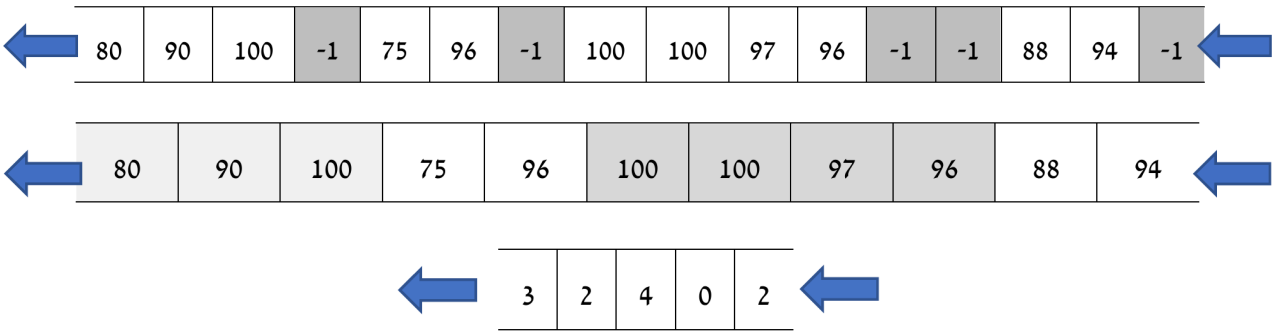
תלמיד חמישי השלים שתי בחינות וקיבל את הציונים הבאים: 88, 94

בבית הספר התקבלה החלטה לשמור את המידע בצורה אחרת, בעזרת שני תורים:

תור ראשון (marks) יכיל ציונים ותור שני (tests) יכיל עבור כל אחד מהתלמידים את מספר הבחינות שהשלים.

תור ציונים marks (אחרי העדכון):

תור בחינות tests:



א. כתבו פעולה בשם arrangeData המקבלת תור ציונים ומחזירה תור חדש המכיל עבור כל אחד מהתלמידים את מספר הבחינות שהשלים. הפעולה גם צריכה לעדכן את תור הציונים כך שיכיל רק ציונים ללא ערך ההפרדה -1.

כותרת הפעולה:

Java:

```
public static Queue<Integer> arrangeData(Queue<Integer> marks)
```

C#:

```
public static Queue<int> ArrangeData(Queue<int> marks)
```

(12 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ב. מהי סיבוכיות הפעולה arrangeData שכתבתם בסעיף א'? **הסבירו את תשובתכם.**

(4 נקודות)

2. שרשרת חוליות המכילה מספרים שלמים (חיוביים, שלילים ואפסים) נקראת "מאוזנת" אם אורכה זוגי וסכום כל הערכים שווה לאפס.

א. כתבו פעולה המקבלת הפניה לחוליה הראשונה של שרשרת חוליות. אם שרשרת "מאוזנת", הפעולה לא עושה דבר. אם לא - הפעולה תוסיף לסוף שרשרת איבר אחד או שני איברים כדי "לאזן" את השרשרת.

כותרת הפעולה:

Java:

```
public static void balance(Node<Integer> chain)
```

C#:

```
public static void Balance(Node<int> chain)
```

(12 נקודות)

ב. מהי סיבוכיות הפעולה balance שכתבתם בסעיף א'? **הסבירו את תשובתכם.**

(4 נקודות)

3.

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

א. נתונה המחלקה Square המתארת ריבוע:

Java:

```
public class Square {
 protected double width;
 protected String color;
 public Square(double val){ width = val; color = "black";}
 public void paint(double scale, String color){..}
 public double perimeter(double scale){ ... }
}
```

C#:

```
public class Square {
 protected double width;
 protected string color;
 public Square(double val){ width = val; color = "black";}
 public void Paint(double scale, string color){..}
 public virtual double Perimeter(double scale){ ... }
}
```

אילו מהחתימות הבאות היא העמסה (overloading) חוקית של הפעולה paint?

הסבירו את תשובתכם!

Java:

```
1. public int paint (double y, String x)
2. public void paint (int x)
3. public double paint (double a, char b)
4. private void paint(double x, String y)
5. private double paint (String y, double x)
```

C#:

```
1. public int Paint (double y, string x)
2. public void Paint (int x)
3. public double Paint (double a, char b)
4. private void Paint(double x, string y)
5. private double Paint (string y, double x)
```

(5 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ב. נתונה המחלקה Rectangle (מלבן) יורשת מ- Square

Java:

```
public class Rectangle extends Square {
  private double length;
  public Rectangle(double val1, double val2){
      super(val1);
      length = val2;
   }
  public boolean equals(Rectangle other){
     return width == other.width && length == other.length;
   }
  }
```

C#:

```
public class Rectangle : Square {
  private double length;
  public Rectangle(double val1, double val2):base(val1)
   {
      length = val2;
   }
  public bool Equals(Rectangle other){
     return width == other.width && length == other.length;
   }
  }
```

עבור כל אחת מבין הפעולות הבאות קבעו אם אפשר להוסיף אותה למחלקה Rectangle?

הסבירו את תשובתכם!

Java:

```
1. private double perimeter (double x){...}
2. public double perimeter (double y) {...}
3. public void perimeter (){...}
4. public int perimeter (int y) {...}
5. public void perimeter (double x) {...}
```

C#:

```
1. private override double Perimeter (double x){...}
2. public override double Perimeter (double y) {...}
3. public override void Perimeter (){...}
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
4. public override int Perimeter (int y) {...}
5. public override void Perimeter (double x) {...}
```

(5 נקודות)

ג. **לפניכם קטע קוד מהתוכנית הראשית:**

Java:

```
Square sqr1 = new Square(1);
Square sqr2 =new Square(1);
Square sqrRec = new Rectangle(1,2);
Rectangle rec = new Rectangle(1,2);
System.out.println(sqr1.equals(sqr2));
System.out.println(sqrRec.equals(rec));
System.out.println(sqrRec.equals((Square)rec));
System.out.println(((Rectangle)sqrRec).equals(rec));
System.out.println(rec.equals(sqrRec));
```

C#:

```
Square sqr1 = new Square(1);
Square sqr2 =new Square(1);
Square sqrRec = new Rectangle(1,2);
Rectangle rec = new Rectangle(1,2);
Console.WriteLine(sqr1.Equals(sqr2));
Console.WriteLine(sqrRec.Equals(rec));
Console.WriteLine(sqrRec.Equals((Square)rec));
Console.WriteLine(((Rectangle)sqrRec).Equals(rec));
Console.WriteLine(rec.Equals(sqrRec));
```

עקבו אחרי ביצוע קטע קוד ורשמו מה יהיה הפלט.

(6 נקודות)

4.

א. נתונה הפעולה main במחלקה Driver:

Java:

```
public static void main(String[] args)
 {
   Three x1 = new One();
   Two x2 = (Two)(new One());
   Two x3 = new Four();
   One x4 = new Four();
 }
```

C#:

```
public static void Main(string[] args)
 {
   Three x1 = new One();
   Two x2 = (Two)(new One());
   Two x3 = new Four();
   One x4 = new Four();
 }
```

נתונות חמש האפשרויות לעץ הירושה הכולל את המחלקות One, Two, Three, Four.

1. Three יורשת מ-One, Two יורשת מ-One, Four יורשת מ-One
2. Three יורשת מ-Two, One יורשת מ-Two, Four יורשת מ-One
3. One יורשת מ-Two, Three יורשת מ-Two, Four יורשת מ-Three
4. One יורשת מ-One, Three יורשת מ-Two, Four יורשת מ-Three
5. One יורשת מ-Two, Three יורשת מ-One, Four יורשת מ-One

עבור כל אחת מהאפשרויות 1-5 ציירו עץ ירושה וכתבו אם הפעולה main תהיה תקינה או תתרחש שגיאת הידור (קומפילציה) או שגיאת זמן ריצה. **הסבירו את תשובתכם.**

(10 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ב. (אין קשר לסעיף א')

נתונות שלוש המחלקות הבאות:

- המחלקה Point מתארת נקודה על ציר ה-X.

Java:

```
public class Point
 {
    private int x;
    public Point (int x)
    {
         this.x = x;
     }
    public String toString()
    {
         return " x= " + this.x;
     }
  }
```

C#:

```
public class Point
 {
    protected int x;
    public Point (int x)
    {
         this.x = x;
     }
    public override string ToString()
    {
         return " x= " + this.x;
     }
  }
```

- המחלקה Circle מתארת מעגל עם מרכז על ציר ה-X

Java:

```
public class Circle extends Point
 {
     private int radius;
     public Circle( int x, int radius)
     {
          ……
     }
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
    public String toString()
    {
        ….
    }
}
```

C#:

```
public class Circle : Point
 {
    private int radius;
    public Circle( int x, int radius): base(….)
    {
        ….
    }
    public override string ToString()
    {
        ….
    }
}
```

- המחלקה Cylinder מייצגת גליל

Java:

```
public class Cylinder extends Circle
 {
    private int height;
    ……..
 }
```

C#:

```
public class Cylinder : Circle
 {
    private int height;
    ……..
 }
```

נתונה הפעולה main הבאה

Java:

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
public static void main(String[] args)
 {
   Circle b = new Circle (1, 2);
   System.out.println(b);
   Cylinder c = new Cylinder (10, 20, 30);
   System.out.println(c);
 }
```

C#:

```
public static void Main(string[] args)
 {
   Circle b = new Circle (1, 2);
   Console.WriteLine(b);
   Cylinder c = new Cylinder (10, 20, 30);
   Console.WriteLine(c);
 }
```

השלימו את הפעולות החסרות במחלקות Cylinder ו- Circle כך שפלט של הפעולה main יהיה

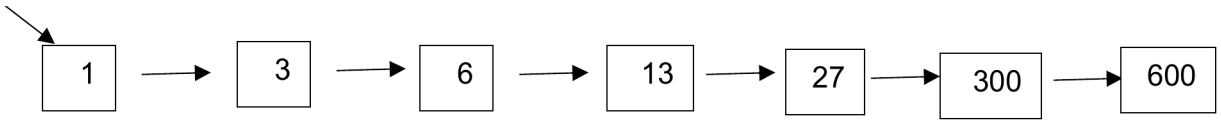
```
x = 1, radius = 2
x = 10, radius = 20, height = 30
```

(6 נקודות)

5. רשימת מספרים שלמים וחיוביים נקראת "סופר עולה" (super-increasing sequence) אם כל איבר ברשימה גדול מסכום של כל האיברים שנמצאים לפניו ברשימה.

לדוגמה:

שרשרת החוליות הבאה מייצגת רשימה "סופר עולה":



נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

א. כתבו פעולה חיצונית בשם isSuper המקבלת הפניה לחוליה ראשונה של שרשרת חוליות ובודקת אם היא מייצגת רשימה "סופר עולה". אם כן - הפעולה תחזיר true, ואם לא - הפעולה תחזיר false.

כותרת הפעולה:

Java:

```
public static boolean isSuper(Node<Integer> n)
```

C#:

```
public static bool IsSuper(Node<int> n)
```

(7 נקודות)

ב. כתבו פעולה המקבלת הפניה לחוליה הראשונה של שרשרת חוליות שהיא רשימה "סופר עולה", ומספר שלם וחיובי num. הפעולה תבדוק האם אפשר להוסיף את המספר num לשרשרת כך שרשימה תישאר "סופר עולה". אם כן - הפעולה תכניס את המספר למיקומו ותחזיר true. אם לא - הפעולה תחזיר false ולא תבצע שום שינוי.

אפשר להניח שמספר num גדול מאיבר הראשון בשרשרת.

לדוגמה: אפשר להוסיף לרשימה כל אחד מהמספרים 60, 90, 1200 שבאיור. לעומת זאת, כל אחד מהמספרים 15, 40, 700, אי אפשר להוסיף לשרשרת שבאיור והיא עדיין תישאר "סופר עולה".

כותרת הפעולה:

Java:

```
public static boolean addToSuper(Node<Integer> n, int num)
```

C#:

```
public static bool AddToSuper(Node<int> n, int num)
```

(7 נקודות)

ג. מהי סיבוכיות הפעולות isSuper ו- addToSuper מהסעיפים א' ו- ב'? **הסבירו את תשובתכם.**

(4 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

6. במכללה מסוימת פותח פרויקט לניהול צוות העובדים. לכל עובד (Employee) במכללה יש שם ומספר עובד. מספר העובד הוא מספר ייחודי הניתן באופן אוטומטי עם הוספת עובד חדש למאגר העובדים כך העובד הראשון יקבל את המספר 1, העובד השני יקבל את המספר 2 וכך הלאה.

למתרגל (Tutor) יש את התכונות:

- num – מספר עובד, מטיפוס שלם, int.
- name – שם, מטיפוס מחרוזת, String.
- courseNum – מספר קורס, מטיפוס שלם, int.

למרצה (Lecturer) יש את התכונות:

- num – מספר עובד, מטיפוס שלם, int.
- name – שם, מטיפוס מחרוזת, String.
- specialization – התמחות (מתמטיקה, מדעי המחשב, אנגלית וכו'), מטיפוס מחרוזת, String.

לראש מחלקה (Head) – ראש מחלקה הוא מרצה שאחראי על צוות שמונה עד עשרים מרצים ומתרגלים, יש את התכונות הבאות:

- arr – מערך של אנשי הצוות הכפופים לראש הצוות.
- current – מספר אנשי הצוות בפועל, מטיפוס שלם, int.

א. סרטטו תרשים UML המתאר את הקשר בין המחלקות Employee, Tutor, Lecturer, Head באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים.

(4 נקודות)

ב. לכל אחת מהמחלקות Employee, Tutor, Lecturer, Head כתבו:

- כותרת המחלקה.
- תכונות.
- פעולה בונה – הפעולה הבונה של כל מחלקה מקבלת את כל הפרמטרים הנדרשים.

> **הערה:**
>
> **יש לשים לב שמספר העובד הוא מספר ייחודי שנוצר באופן אוטומטי ואין להעבירו כפרמטר לפעולה בונה!**

(6 נקודות)

ג. המחלקה TeachingStaff מיועדת לנהל את כל העובדים במכללה. תכונות המחלקה:

- arr – מערך של כל אנשי הצוות. במכללה יכולים לעבוד לא יותר מ- 200 אנשי צוות.
- current – מספר אנשי הצוות בפועל, מטיפוס שלם, int.

כתבו את כותרת המחלקה ואת התכונות של המחלקה TeachingStaff והוסיפו את הפעולות הפנימיות הבאות:

1. פעולה בשם numHeads המחזירה את מספר הצוותים.
2. פעולה בשם getNewTutor המקבלת את מספר הקורס ומחזירה את המתרגל המלמד בקורס שהצטרף אחרון למאגר העובדים (מספר העובד שלו מקסימלי). אם אין מתרגל בקורס הנדרש, הפעולה תחזיר null.

הערה: הניחו שהפעולות get ו- set מוגדרות בעבור כל תכונה בכל אחת מהמחלקות.

(8 נקודות)

7. לפניכם הפעולות הבאות:

Java:

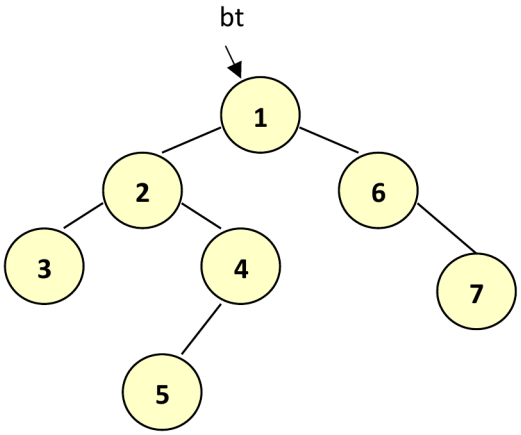
```
public static void what(Stack<Integer> s)
{
  if (!s.isEmpty())
   {
    System.out.print(s.pop()+" ");
    what(s);
   }
}
public static void doWhat(BinNode<Integer> t, int data)
 {
    Stack<Integer>s = new Stack<Integer>();
    doWhat(t, s, data);
 }
private static void doWhat(BinNode<Integer> t, Stack<Integer> s,
                    int val)
{
   if (t!=null)
    {
     int x = t.getValue();
     s.push(x);
     if(x == val)
     {
      what(s);
     }
     else
     {
      doWhat(t.getLeft(), s, val);
          if (!s.isEmpty())
            doWhat(t.getRight(), s, val);
          if (!s.isEmpty())
           s.pop();
       }
    }
 }
```

C#:

```
public static void What(Stack<int> s)
{
  if (!s.IsEmpty())
   {
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
    Console.Write(s.Pop()+" ");
    What(s);
   }
}
public static void DoWhat(BinNode<int> t, int data)
{
    Stack<int>s = new Stack<int>();
    DoWhat(t, s, data);
 }
private static void DoWhat(BinNode<int> t, Stack<int> s,
                    int val)
{
  if (t!=null)
   {
    int x = t.GetValue();
    s.Push(x);
    if(x == val)
    {
     What(s);
    }
    else
    {
     DoWhat(t.GetLeft(), s, val);
        if (!s.IsEmpty())
         DoWhat(t.GetRight(), s, val);
        if (!s.IsEmpty())
         s.Pop();
     }
   }
}
```



א. מה מבצעת הפעולה what(Stack<Integer> s) עבור מחסנית של מספרים שלמים s?

(2 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ב. עקבו אחרי זימון הפעולה doWhat(bt,3) עבור עץ bt שבאיור וכתבו מה יהיה הפלט של הזימון.

(6 נקודות)

ג. עקבו אחרי זימון הפעולה doWhat(bt,5) עבור עץ bt שבאיור וכתבו מה יהיה הפלט של הזימון.

(6 נקודות)

ד. מה יהיה פלט של זימון הפעולה doWhat(bt,10) עבור עץ bt שבאיור?

(2 נקודות)

ה. מה מבצעת הפעולה doWhat(t, val) עבור עץ בינרי t ומספר שלם val באופן כללי?

(2 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

8. נתונות שלוש המחלקות הבאות:

Java:

```
public class A
{
  public A()
  {
    System.out.println("A");
  }

  public void one()
  {
    System.out.println("one of A");
  }

  public void two()
  {
    System.out.println("two of A");
    this.one();
  }

  public void four()
  {
    System.out.println ("four of A");
    this.two();
  }
} //end of class A

public class B extends A
{
  public B()
  {
    System.out.println("B");
  }

  public void one()
  {
    System.out.println("one of B");
  }

  public void two()
  {
    System.out.println("two of B");
  }

  public void three()
```

```
  {
    System.out.println("three of B");
    super.two();
  }
} //end of class B

public class C extends B
{
  public void four()
  {
    System.out.println ("four of C");
  }
} //end of class C
```

C#:

```
public class A
{
  public A()
  {
    Console.WriteLine("A");
  }

   public virtual void One()
   {
     Console.WriteLine("One of A");
   }

   public virtual void Two()
   {
     Console.WriteLine("Two of A");
     this.One();
   }

  public virtual void Four()
  {
    Console.WriteLine ("Four of A");
    this.Two();
  }
} //end of class A

public class B : A
{
  public B()
  {
```

```
    Console.WriteLine("B");
  }

  public override void One()
  {
    Console.WriteLine("One of B");
  }

  public override void Two()
  {
    Console.WriteLine("Two of B");
  }

  public void Three()
  {
    Console.WriteLine("Three of B");
    base.Two();
  }
} //end of class B

public class C : B
{
  public override void Four()
  {
    Console.WriteLine ("Four of C");
  }
} //end of class C
```

כמו כן נתונה המחלקה Test הבאה באותו פרויקט:

Java:

```
public class Test
{
  public static void main (String [] args)
  {

    // כאן יופיעו הפקודות שבסעיפים להלן

  }
}
```

C#:

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
public class Test
{
  public static void Main(string [] args)
  {

    // כאן יופיעו הפקודות שבסעיפים להלן

  }
}
```

בכל סעיף עליכם לכתוב מה יקרה בשל הרצת קטע הקוד. אם תהיה שגיאת קומפילציה או הרצה, כתבו איזו שגיאה ונמקו. אם הקוד תקין, כתבו מה יודפס בפלט.

א. Java:

```
A a1 = new A();
((B) a1).two();
```

C#:

```
A a1 = new A();
((B) a1).Two();
```

(3 נקודות)

ב. Java:

```
B b1 = new B();
((A) b1).three();
```

C#:

```
B b1 = new B();
((A) b1).Three();
```

(3 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ג. Java:

```
A a2 = new A();
a2.one();
((A) a2).two();
```

C#:

```
A a2 = new A();
a2.One();
((A) a2).Two();
```

(3 נקודות)

ד. Java:

```
A a3 = new B();
a3.two();
((B) a3).two();
((B) a3).three();
```

C#:

```
A a3 = new B();
a3.Two();
((B) a3).Two();
((B) a3).Three();
```

(3 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ה. Java:

```
B b2 = new B();
b2.two();
((A) b2).two();
```

C#:

```
B b2 = new B();
b2.Two();
((A) b2).Two();
```

(3 נקודות)

ו. Java:

```
B b3 = new C();
b3.four();
((B)new C()).four();
```

C#:

```
B b3 = new C();
b3.Four();
((B)new C()).Four();
```

(3 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

9. נתונה הפעולה הרקורסיבית הבאה:

Java:

```
public static int secret(Stack<Integer> s1, Stack<Integer> s2)
{
    if(s1.isEmpty() || s2.isEmpty())
        return 0;
    if(s1.top() > s2.top())
        return s1.pop() + s2.top() + secret(s1, s2);
    else
        return s1.top() + s2.pop() + secret(s1, s2);
}
```

C#:

```
public static int Secret(Stack<int> s1, Stack<int> s2)
{
    if(s1.IsEmpty() || s2.IsEmpty())
        return 0;
    if(s1.Top() > s2.Top())
        return s1.Pop() + s2.Top() + Secret(s1, s2);
    else
        return s1.Top() + s2.Pop() + Secret(s1, s2);
}
```



א. נתונות שתי המחסניות הבאות:

עקבו אחרי זימון הפעולה secret(st1, st2) וכתבו מה תחזיר הפעולה.

יש להראות שינוים בתכנים של המחסניות.

(8 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ב. נתונה הטענה הבאה:

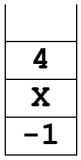
```
secret(s1, s2) == secret(s2, s1)
```

האם הטענה נכונה לכל זוג מחסניות של מספרים שלמים s1 ו- s2? **נמקו את תשובתכם.**

אם הטענה לא נכונה, יש להביא דוגמה של זוג מחסניות s1 ו- s2 מתאים.

(4 נקודות)

s1



s2



ג. נתונות שתי המחסניות הבאות:

האם קיים ערך X כזה שתוצאת זימון הפעולה secret(s1, s2) תהיה 19?

הסבירו את תשובתכם.

(4 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

10. הגדרה:

קבוצת מספרים - SetOfNumbers - היא אוסף של מספרים, כאשר כל מספר יכול להופיע בקבוצה פעם אחת בלבד. **אין חשיבות לסדר המספרים בקבוצה.**

לפניכם ממשק חלקי של המחלקה SetOfNumbers המייצגת קבוצת מספרים **שלמים**.

כותרת הפעולה	תיאור הפעולה
`public SetOfNumbers()`	הפעולה בונה קבוצת מספרים ריקה
`public void addToSet(int num)` (C#: `public void AddToSet(int num)`)	הפעולה מוסיפה לקבוצה את המספר num, אם המספר לא נמצא בקבוצה. אם num כבר נמצא בקבוצה, הפעולה לא עושה דבר
`public void removeFromSet(int num)` (C#: `public void RemoveFromSet(int num)`)	הפעולה מוחקת מהקבוצה את המספר num, אם הוא נמצא בקבוצה. אם num לא נמצא בקבוצה, הפעולה לא עושה דבר
`public int getMax()` (C#: `public int GetMax()`)	הפעולה מחזירה את מספר הגדול בקבוצה. **הנחה**: הקבוצה לא ריקה
`public boolean isEmpty()` (C#: `public bool IsEmpty()`)	הפעולה מחזירה true אם הקבוצה הנוכחית ריקה, אחרת - הפעולה מחזירה false

שמו לב! מימוש של המחלקה SetOfNumbers לא ידוע, אפשר להשתמש אך ורק בפעולות הנתונות בממשק!

א. כתבו במחלקה SetOfNumbers פעולה פנימית המחזירה את המספר הקטן בקבוצה.

הנחה: הקבוצה לא ריקה.

כותרת הפעולה:

Java:

```
public int getMin()
```

C#:

```
public int GetMin()
```

(6 נקודות)

ב. כתבו במחלקה SetOfNumbers פעולה פנימית המקבלת קבוצה אחרת. הפעולה תחזיר true, אם שתי קבוצות זהות, אם לא - הפעולה תחזיר false.

כותרת הפעולה:

Java:

```
public boolean equals(SetOfNumbers other)
```

C#:

```
public bool Equals(SetOfNumbers other)
```

(6 נקודות)

ג. כתבו פעולה חיצונית שמקבלת קבוצה של מספרים שלמים sn ומחזירה קבוצה חדשה שתכיל רק את המספרים הדו-ספרתיים שנמצאים ב-sn. **יש לשמור על המספרים שנמצאים בקבוצה sn**

כותרת הפעולה:

Java:

```
public static SetOfNumbers make(SetOfNumbers sn)
```

C#:

```
public static SetOfNumbers Make(SetOfNumbers sn)
```

(4 נקודות)

כל מבחני מה"ט עם פתרונות מלאים: www.guyai.co.il/mahat