GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: קיץ 2023 מועד א'
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 16) 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.**

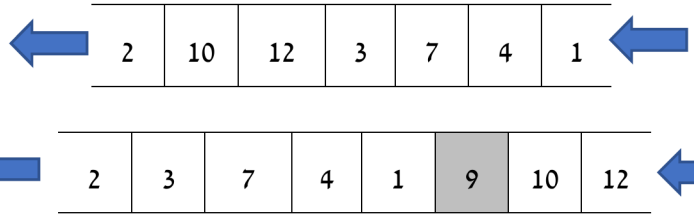
ד. הוראות מיוחדות: 1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות הבחינה, ולוודא שהן מובנות. 2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי (גם בכך תלויה הערכת הבחינה). 3. יש להשאיר את העמוד הראשון במחברת הבחינה ריק. בסיום המבחן יש לרשום בעמוד זה את מספרי התשובות לבדיקה. התשובות ייבדקו לפי 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.



א. כתבו פעולה `putInPlace` המקבלת תור `q` ומספר `num` וממקמת את המספר `num` במיקומו, כך שכל האיברים הקטנים ממנו, נמצאים לפניו וכל האיברים השווים לו או הגדולים ממנו נמצאים אחריו.

(אין משמעות לסדר של האיברים הקטנים או הגדולים ממנו).

הפעולה תחזיר את מיקומו הסידורי של המספר `num` בתוך התור אחרי הכנסתו.

לדוגמה:

עבור התור `q` הבא: 2, 10, 12, 3, 7, 4, 1 (2 בראש התור)

והמספר `num=9`

התור שיתקבל יכול להיות: 2, 3, 7, 4, 1, 9, 10, 12

והפעולה תחזיר **6** (מיקומו של ה-`num` אחרי הכנסתו).

כותרת הפעולה:

Java:

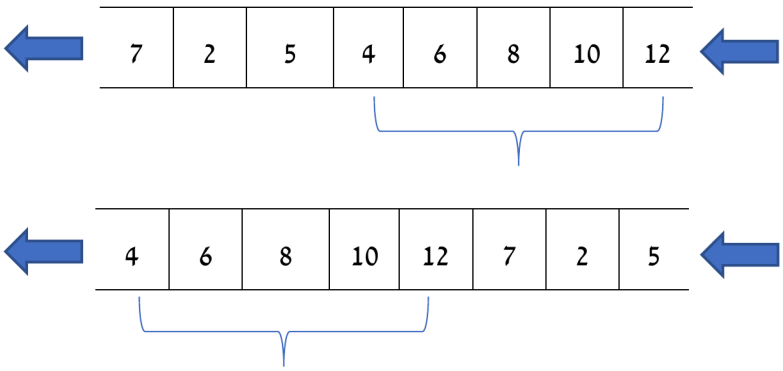
```
public static int putInPlace(Queue<Integer> q, int num)
```

C#:

```
public static int PutInPlace(Queue<int> q, int num)
```

(6 נקודות)

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



ב. כתבו פעולה חיצונית המקבלת תור `q` של מספרים שלמים ומספר שלם וחיובי `k`, ומעבירה את `k` **האיברים האחרונים בתור אל ראש התור**. אפשר להניח ש-`k` קטן מאורך התור.

לדוגמה:

עבור התור `q` הבאה: 7, 2, 5, 4, 6, 8, 10, 12 (7 בראש התור)

והמספר `k=5` התור `q` יראה כך: 4, 6, 8, 10, 12, 7, 2, 5

כותרת הפעולה:

Java:

```
public static void moveToFront(Queue<Integer> q, int k)
```

C#:

```
public static void MoveToFront(Queue<int> q, int k)
```

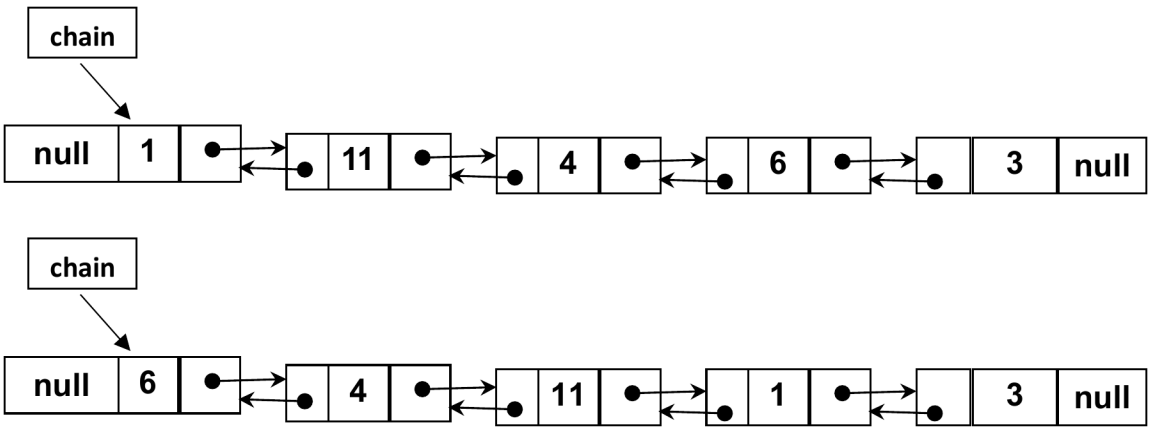
(6 נקודות)

ג. מהי הסיבוכיות של הפעולות שכתבתם בסעיפים א' ו-ב'? **הסבירו את תשובתכם.**

(4 נקודות)

2. בעזרת המחלקה `BinNode` אפשר לממש שרשרת חוליות דו-כיוונית.

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



א. כתבו פעולה המקבלת הפנייה לחוליה הראשונה (הכי שמאלית) של שרשרת חוליות הדו-כיוונית ומסדרת אותה כך שכל הערכים הזוגיים יהיו בתחילת השרשרת וכל הערכים האי-זוגיים יהיו בסופה.

סדר האיברים לא חשוב.

לדוגמה:

אם לפני הפעולה השרשרת הייתה: 1, 11, 4, 6, 3

אז אחרי הפעולה היא __יכולה להיות__: 6, 4, 11, 1, 3

שימו לב: אין להשתמש במבנה נתונים נוסף!

כותרת הפעולה:

Java:

```
public static void order(BinNode<Integer> chain)
```

C#:

```
public static void Order(BinNode<int> chain)
```

(12 נקודות)

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

(4 נקודות)

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

3. נתונות שתי המחלקות `Flower` ו- `Rose`:

Java:

```
public class Flower
{
 protected int height;
 private int price;
 public Flower (int val) { this.height = val; this.price = 10;}
 public int getHeight() { return this.height; }
 public int getPrice () { return this.price; }
}
public class Rose extends Flower
{
 private String color;
 public Rose(int val, String col)
  {
     super(val);
     this.color=col;
  }
 public boolean validHeight()
  {
    return this.height > 10 && this.height < 30;
  }
}
```

C#:

```
public class Flower
{
 protected int height;
 private int price;
 public Flower (int val) { this.height = val; this.price = 10;}
 public int GetHeight() { return this.height; }
 public int GetPrice () { return this.price; }
}
public class Rose : Flower
{
 private string color;
 public Rose(int val, string col): base(val)
  {
     this.color=col;
  }
 public bool ValidHeight()
  {
```

```
    return this.height > 10 && this.height < 30;
  }
}
```

א. לפניכם חמישה היגדים. קבעו לכל אחד מהם אם הוא נכון או אינו נכון, ונמקו את קביעתכם.

1. המחלקה **Flower** יורשת את הפעולה `validHeight()` מהמחלקה **Rose**.
2. המחלקה **Rose** יורשת את כל התכונות ואת כל הפעולות של המחלקה **Flower**.
3. המחלקה **Rose** יכולה לגשת ישירות לתכונה `height` של המחלקה **Flower**.
4. המחלקה **Flower** יכולה לגשת לתכונה `color` של המחלקה **Rose**.
5. לעצמים מטיפוס **Rose** אין תכונה `price`.

(5 נקודות)

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

ב. לפניכם קטע קוד מהתוכנית הראשית:

Java:

```
Flower  first = new Rose (20, "RED");
Flower  second = new Flower (93);
```

C#:

```
Flower  first = new Rose (20, "RED");
Flower  second = new Flower (93);
```

בעבור כל אחת מההוראות שלפניך קבעו אם היא תקינה או אינה תקינה.

אם היא אינה תקינה, נמקו את קביעתכם וכתבו אם זו שגיאת ריצה או שגיאת הידור (קומפילציה).

Java:

```
1. boolean b = first. validHeight();
2. boolean b = second. validHeight();
3. boolean b = ((Rose)first). validHeight();
4. boolean b = ((Rose)second). validHeight();
5. boolean b = first.getPrice() == second.price;
```

C#:

```
1. bool b = first. ValidHeight();
2. bool b = second. ValidHeight();
3. bool b = ((Rose)first). ValidHeight();
4. bool b = ((Rose)second). ValidHeight();
5. bool b = first.GetPrice() == second.price;
```

(5 נקודות)

ג. כתבו פעולה המקבלת מערך עצמים מטיפוס `Object`. על הפעולה להדפיס כמה עצמים הם מטיפוס **Rose**, כמה עצמים מטיפוס **Flower** ואינו מטיפוס **Rose** וכמה עצמים הם לא מטיפוס **Flower**.

(6 נקודות)

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

4. בפרויקט מסוים הוגדרו שלשה ממשקים ושתי מחלקות:

Java:

```
public interface Dancer{
   void lonelyDance();
   void coupleDance(Dancer d);
}
public interface Singer   {   void sing();  }
public interface Musician {   void play();  }
public class Actor implements Dancer, Singer, Musician
{
   private String name;
   private int age;
   ...
}
public class Driver
{
   public static void main(String[] args)
   {
      Actor actor1 = new Actor("Bob", 25);
      Actor actor2 = new Actor("Alice");
      ( ·····)
   }
}
```

C#:

```
public interface Dancer{
    void LonelyDance();
    void CoupleDance(Dancer d);
}
public interface Singer {     void Sing(); }
public interface Musician {   void Play(); }
public class Actor : Dancer, Singer, Musician
{
    private string name;
    private int age;
    ...
}
public class Driver
{
    public static void Main(string[] args)
    {
        Actor actor1 = new Actor("Bob", 25);
```


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

```
        Actor actor2 = new Actor("Alice");
        ( ·····)
    }
}
```

א. השלימו את המחלקה `Actor`: תכתבו כותרות של הפעולות החסרות בה.

(4 נקודות)

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

ב. בכל אחד מסעיפים הבאים השורה `( *)` תוחלף בשורת קוד משורות הקוד 1-6. כתבו, עבור כל אחד מהסעיפים, אם הקוד תקין או לא. אם הקוד אינו תקין - יש להסביר למה הוא אינו תקין ולציין את סוג השגיאה (קומפילציה או זמן ריצה).

שימו לב שאין קשר בין סעיפים!

Java:

```
(1)  Dancer dancer1 = actor1;
     dancer1. coupleDance (actor2);

(2)  Singer singer2 = new Singer();
     singer2.sing();

(3)  Singer singer3 = new Actor ("Clark",30);
     Dancer dancer3 = (Actor)singer3;

(4)  Singer actor4 = new Actor("Danny");
     actor4.play();

(5)  Dancer actor5 = new Actor("Eddie");
     Singer singer5 = (Singer) actor5 ;
     singer5.sing();

(6)  Musician musician6 = new Actor("Freddie", 45);
     ((Actor) musician6).lonelyDance();
     ((Singer) musician6).sing();
```

C#:

```
(1)  Dancer dancer1 = actor1;
     dancer1. CoupleDance (actor2);

(2)  Singer singer2 = new Singer();
     singer2.Sing();

(3)  Singer singer3 = new Actor ("Clark",30);
     Dancer dancer3 = (Actor)singer3;

(4)  Singer actor4 = new Actor("Danny");
     actor4.Play();

(5)  Dancer actor5 = new Actor("Eddie");
     Singer singer5 = (Singer) actor5 ;
     singer5.Sing();
```


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

```
(6)  Musician musician6 = new Actor("Freddie", 45);
     ((Actor) musician6).LonelyDance();
     ((Singer) musician6).Sing();
```

(6 נקודות)

ג. (אין קשר לסעיפים א' ו-ב')

נתונה הפעולה `main` הבאה:

Java:

```
public static void main(String[] args)
{
  C c = new A();
  B b1 = (B)(new A());
  B b2 = new D();
  A a = new D();
}
```

C#:

```
public static void Main(string[] args)
{
  C c = new A();
  B b1 = (B)(new A());
  B b2 = new D();
  A a = new D();
}
```

עבור כל אחת מן האפשרויות 1-6, קבעו אם יחס ההורשה בין המחלקות מתאים לקוד של הפעולה `main` הנתונה.

הסבירו את תשובתכם.

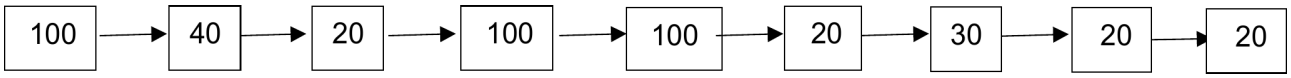
1. C יורשת מ- A, B יורשת מ- A, D יורשת מ- A.
2. A יורשת מ- C, B יורשת מ- A, D יורשת מ- A.
3. C יורשת מ- A, B יורשת מ- A, D יורשת מ- B.
4. A יורשת מ- C, C יורשת מ- B, D יורשת מ- A.
5. A יורשת מ- C, B יורשת מ- A, D יורשת מ- B.
6. A יורשת מ- C, B יורשת מ- C, D יורשת מ- C.

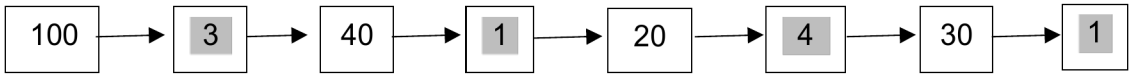
(6 נקודות)

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

5. נתונה רשימה של מספרים שלמים. כל איבר (מספר) ברשימה יכול להופיע פעם אחת או יותר.

רשימת תדירויות (Frequency List) היא רשימה של מספרים שלמים שבה כל מספר מופיע __פעם אחת__ ולאחריו מספר הפעמים שהופיע ברשימה המקורית.





א. כתבו פעולה המקבלת רשימה של מספרים שלמים (הפנייה לחוליה ראשונה של הרשימה), ברשימה יש לפחות ערך אחד. הפעולה תשנה את הרשימה להיות **רשימת תדירויות**.

שימו לב: הפעולה __תעדכן__ את הרשימה שהתקבלה כך שכל ערך יופיע פעם אחת ולאחריו מספר הפעמים שהופיע ברשימה המקורית.

יש למחוק ערכים שחוזרים על עצמם חוץ מהמופע הראשון.

אין להשתמש במבני נתונים נוספים!!!

לדוגמה:

הרשימה המקורית: 100, 40, 20, 100, 100, 20, 30, 20, 20

- מספר 100 מופיע שלוש פעמים.
- מספר 40 מופיע פעם אחת.
- מספר 20 מופיע ארבע פעמים.
- מספר 30 מופיע פעם אחת.

הרשימה אחרי העדכון (רשימת תדירויות): 100, 3, 40, 1, 20, 4, 30, 1

כותרת הפעולה:

Java:

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

C#:

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

(7 נקודות)

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

ב. כתבו פעולה המקבלת הפניה לחוליה ראשונה של שרשרת חוליות ומחזירה את המספר __המופיע הכי הרבה__ פעמים בשרשרת. בעבור הרשימה הנתונה בסעיף א' הפעולה תחזיר 20. כותרת הפעולה:

Java:

```
public static int mostPopularNumber(Node<Integer> chain)
```

C#:

```
public static int MostPopularNumber(Node<int> chain)
```

חובה להשתמש בפעולה שכתבתם בסעיף א'.

(7 נקודות)

ג. מהי סיבוכיות הפעולות `buildFreqList` ו-`mostPopularNumber` מהסעיפים א' ו-ב'? **הסבירו את תשובתכם.**

(4 נקודות)

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

6. במלון "בא ונח" אפשר להזמין שלושה סוגים שונים של חדרים - חדר רגיל, חדר משפחתי וסוויטה.
מערכת ניהול המלון כוללת ארבע מחלקות: `Hotel`, `Suite`, `FamilyRoom`, `Room`.
לחדר רגיל (`Room`) יש את התכונות הבאות:
- `num` - מספר חדר, מטיפוס שלם, `int`
- `beds` - מספר המיטות, מטיפוס שלם, `int`
- `price` - מחיר, מטיפוס ממשי, `double`

לחדר משפחה (`FamilyRoom`) יש את התכונות הבאות:
- `num` - מספר חדר, מטיפוס שלם, `int`
- `beds` - מספר המיטות, מטיפוס שלם, `int`
- `price` - מחיר, מטיפוס ממשי, `double`
- `babyBed` - האם נוספה מיטת תינוק, מטיפוס `boolean` (`true` - כן נוספה מיטה, `false` - לא)
- `balcony` - האם יש מרפסת, מטיפוס `boolean` (`true` - יש, `false` - אין)

לסוויטה (`Suite`) יש את התכונות הבאות:
- `num` - מספר חדר, מטיפוס שלם, `int`
- `beds` - מספר המיטות, מטיפוס שלם, `int`
- `price` - מחיר, מטיפוס ממשי, `double`
- `jacuzzi` - האם יש ג'קוזי, מטיפוס `boolean` (`true` - יש, `false` - אין)

א. סרטטו תרשים UML המתאר את הקשר בין המחלקות: `Suite`, `FamilyRoom`, `Room` באופן __המתאים ביותר לעקרונות של תכנות מונחה עצמים__.
(2 נקודות)

ב. לכל אחת מהמחלקות `Suite`, `FamilyRoom`, `Room` כתבו כותרת המחלקה ותכונות שלה.
אפשר להניח שבכל מחלקה הוגדרו פעולות `set/get`
(3 נקודות)

ג. לאחר עזיבת האורחים יש להחליף מצעים בכל אחת מהמיטות שבחדר, לשטוף את הריצפה ולנקות אבק בחדר ובמרפסת, להוציא מיטת תינוק בחדר משפחה (אם יש), לנקות את הג'קוזי בסוויטה (אם יש) ולשאוב שטיחים בסוויטה.

הזמן הדרוש לכל אחד מסוגי עבודת ניקיון הוא:

- החלפת מצעים למיטה אחת - חמש דקות,
- שטיפת ריצפה וניקוי אבק - עשר דקות בחדר רגיל וחדר משפחה, עשרים דקות בסוויטה
- ניקוי מרפסת - חמש דקות
- קיפול והוצאת מיטת תינוק - חמש דקות
- ניקוי ג'קוזי - עשרים דקות
- שאיבת שטיחים - חמש עשרה דקות

כתבו פעולה או פעולות לחישוב ולהחזרה של הזמן הדרוש לניקיון חדר.

יש לציין באיזו מחלקה צריך להוסיף את הפעולה או את הפעולות.

(6 נקודות)

ד. למחלקה `Hotel` (מלון) יש תכונה אחת - מערך של חדרי המלון.

כתבו את כותרת המחלקה והגדירו את התכונה.

(1 נקודות)

ה. בסוף כל יום עבודה האחראי על הניקיון מכין תוכנית העבודה ליום המחרת. לשם כך הוא מקבל רשימה של __מספרי חדרים__ שיתפנו למחרת.

כתבו במחלקה `Hotel` פעולה פנימית המקבלת מערך של מספרים שלמים (מערך מספרי חדרים).

הפעולה תחשב ותחזיר את הזמן הכולל של הניקיון. הפעולה גם תדפיס את מספר מיטות התינוק שיתפנו.

כותרת הפעולה:

Java:

```
public int totalTime(int[] numArr)
```

C#:

```
public int TotalTime(int[] numArr)
```

(6 נקודות)

7.

א. לפניכם פעולה רקורסיבית המקבלת הפנייה לשורש של עץ בינרי:

Java:

```
public static  boolean special(BinNode<Integer> t)
{
   return special(t, 0);
}
private static  boolean special(BinNode<Integer> t, int r)
{
  if (t==null)
      return true;
  if (t.getValue()<r)
      return false;
 return special(t.getLeft(), r+1) && special(t.getRight(), r+1);
}
```

C#:

```
public static  bool Special(BinNode<int> t)
{
   return Special(t, 0);
}
private static  bool Special(BinNode<int> t, int r)
{
  if (t==null)
      return true;
  if (t.GetValue()<r)
      return false;
 return Special(t.GetLeft(), r+1) && Special(t.GetRight(), r+1);
}
```

תנו דוגמה לעץ המכיל לפחות שישה איברים שעבורו תחזיר הפעולה `special(t)` את הערך `true`.

(5 נקודות)

ב. מהי מטרת הפעולה `special`?

(4 נקודות)

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

ג. נתונות שתי פעולות המקבלות הפנייה לשורש של **עץ חיפוש בינרי** (Binary Search Tree):

Java:

```
public static int what(BinNode<Integer> bt)
{
  if(bt.getLeft() == null)
    return bt.getValue();
  return what(bt.getLeft());
}

public static int where(BinNode<Integer> bt)
{
  while(bt.getRight()!=null)
  {
    bt = bt.getRight();
  }
  return bt.getValue();
}
```

C#:

```
public static int What(BinNode<int> bt)
{
  if(bt.GetLeft() == null)
    return bt.GetValue();
  return What(bt.GetLeft());
}

public static int Where(BinNode<int> bt)
{
  while(bt.GetRight()!=null)
  {
    bt = bt.GetRight();
  }
  return bt.GetValue();
}
```

מהן מטרות הפעולות `what` ו-`where`?

(4 נקודות)

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

ד. האם קיים **עץ חיפוש בינרי** `bt` הכולל לפחות שלושה צמתים שעבורו מתקיים:

`what(bt) = where(bt)`?

אם כן - ציירו את העץ ולא - הסבירו מדוע עץ כזה לא קיים.

(5 נקודות)

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

8. נתונות שתי המחלקות הבאות:

Java:

```
public class A {
   public static int countA = 0;
   private String st;
   public A() { this.st = "aaa"; countA++;}
   public A(String st) { this.st = st; countA++;}
   public String toString() { return this.st; }
   public boolean same (Object obj) {
      if ((obj instanceof A) && (((A)obj).st.equals(this.st)))
         return true;
      return false;
   }
}   //end of class A

public class B extends A {
   public static int countB = 0;
   private int num;
   public B() {
      super();
      this.num = 1;
      countB++;
   }
   public B(int num) {
      super();
      this.num = num;
      countB++;
   }
   public B(int num, String st) {
      super(st);
      this.num = Math.abs(num);
      countB++;
   }
   public String toString() {
      String s = this.num +"/";
      for(int i = 0; i< this.num; i++)
         s+=super.toString();
      return s;
   }
   public boolean same (Object obj) {
      if ((obj instanceof B) && (((B)obj).num == this.num))
         return true;
      return false;
```


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

```
    }
} //end of class B
```

C#:

```
using System;

public class A {
   public static int countA = 0;
   private string st;
   public A() { this.st = "aaa"; countA++;}
   public A(string st) { this.st = st; countA++;}
   public override string ToString() { return this.st; }
   public virtual bool Same (Object obj) {
      if ((obj is A) && (((A)obj).st.Equals(this.st)))
         return true;
      return false;
   }
}   //end of class A

public class B : A {
   public static int countB = 0;
   private int num;
   public B(): base() {
      this.num = 1;
      countB++;
   }
   public B(int num): base() {
      this.num = num;
      countB++;
   }
   public B(int num, string st): base(st) {
      this.num = Math.Abs(num);
      countB++;
   }
   public override string ToString() {
      string s = this.num +"/";
      for(int i = 0; i< this.num; i++)
         s+= base.ToString();
      return s;
   }
   public override bool Same (Object obj) {
      if ((obj is B) && (((B)obj).num == this.num))
         return true;
      return false;
```

```
  }
} //end of class B
```

א. לפניכם קטע קוד מהתוכנית הראשית:

Java:

```
A a = new A();
B b = new B();
if (a.same (b))
   System.out.println (a);
else
   System.out.println (b);
if (b.same(a))
   System.out.println (b);
else
   System.out.println (a);
```

C#:

```
A a = new A();
B b = new B();
if (a.Same(b))
  Console.WriteLine (a);
else
  Console.WriteLine (b);
if (b.Same(a))
   Console.WriteLine (b);
else
  Console.WriteLine (a);
```

מה יהיה הפלט של קטע התוכנית? הסבירו את תשובתכם.

(4 נקודות)

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

ב. נתונה המחלקה `Tester` הבאה:

Java:

```
public class Tester {
   public static void print (Object[] arr)
   {
      for (int i = 0; i < arr.length; i++)
      {
         if (arr[i] != null)
         {
            if (arr[i] instanceof B)
               System.out.println("B: " + arr[i]);
            else if (arr[i] instanceof A)
                 System.out.println("A: " + arr[i]);
               else
                 System.out.println("Unknown");
         }
         else
          System.out.println("NULL");
      }
   }
   public static void main (String[] args)
   {
      A[] ar1 = new A[7];
      ar1[0] = new B(-4);
      ar1[1] = new A("bbb");
      ar1[2] = new B(3,"ccc");
      ar1[3] = new B();
      ar1[4] = new B(2, "ddd");
      ar1[5] = new A();
      print(ar1);
      System.out.println(A.countA+ " "+ B.countB);
   }
} //end of class Tester
```

C#:

```
using System;

public class Tester {
    public static void Print (Object[] arr)
    {
        for (int i = 0; i < arr.Length; i++)
        {
```


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

```
        if (arr[i] != null)
        {
            if (arr[i] is B)
                Console.WriteLine("B: " + arr[i]);
            else if (arr[i] is A)
                  Console.WriteLine("A: " + arr[i]);
                else
                  Console.WriteLine("Unknown");
        }
        else
         Console.WriteLine("NULL");
      }
   }
   public static void Main (String[] args)
   {
      A[] ar1 = new A[7];
      ar1[0] = new B(-4);
      ar1[1] = new A("bbb");
      ar1[2] = new B(3,"ccc");
      ar1[3] = new B();
      ar1[4] = new B(2, "ddd");
      ar1[5] = new A();
      Print(ar1);
      Console.WriteLine(A.countA+ " "+ B.countB);
   }
} //end of class Tester
```

עקבו אחרי ביצוע תוכנית הראשית (`main`) ורשמו מה יהיה פלט התוכנית. יש להציג את התכונות של כל העצמים שנוצרו במהלך ביצוע התוכנית.

(14 נקודות)

9.

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

א. נתונה הפעולה הרקורסיבית `what` המקבלת תור של מספרים `q` ומספר `x`:

Java:

```
public  static  int  what (Queue<Integer> q , int x)
{
 if ( q.isEmpty() || x != q.head() )
       return 0;
   x = q.remove();
   return   1 + what(q , x);
}
```

C#:

```
public static int What (Queue<int> q , int x)
{
 if ( q.IsEmpty() || x != q.Head() )
       return 0;
   x = q.Remove();
   return   1 + What(q , x);
}
```

(1) מה יחזיר זימון הפעולה `what(q,5)` עבור התור `q:[5,5,5,3,3,2,8]` (5 בראש התור)?

__יש להראות מעקב אחרי ביצוע הפעולה.__

(2) אחרי זימון הפעולה `what(q, x)` עבור התור `q` לא ריק, התור `q` נשאר ללא שינוי. האם הטענה "ערך `x` לא מופיע בתור" נכונה? הסבירו את תשובתכם.

(3) תנו דוגמה של תור `q` שעבורו זימון הפעולה `what(q, 7)` תחזיר 4. איך ייראה התור אחרי זימון הפעולה?

(4) מה מבצעת הפעולה ?

(8 נקודות)

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

ב. נתונה הפעולה `secret` המקבלת תור של מספרים `q`:

Java:

```
public static int secret(Queue<Integer> q)
{
 if (q.isEmpty()) return 0;
 int x = q.head();
 int z = x * what(q, x);
 int y = 1 + secret(q);
 q.insert(z);
 return y;
}
```

C#:

```
public static int Secret(Queue<int> q)
{
 if (q.IsEmpty()) return 0;
 int x = q.Head();
 int z = x * What(q, x);
 int y = 1 + Secret(q);
 q.Insert(z);
 return y;
}
```

עקבו אחר הפעולה `secret(q)` עבור התור `q : [5,5,1,1,4,4,4,5,6]` (5 בראש התור).

אין צורך לפרט את המעקב אחרי הפעולה `what`.

(1) מהו הערך המוחזר מהפעולה `secret`?

(2) איך ייראה התור בסיום הפעולה `secret`?

(3) __אחרי__ זימון הפעולה `secret(q)` התור `q` נראה כך: `q:[6,16,2,3]` (6 בראש התור)
איך נראה התור `q` __לפני__ זימון הפעולה?

(8 נקודות)

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

10. המחלקה `Card` מייצגת קלף למשחק ילדים Eka.

במשחק משתתפים קלפים בארבעה צבעים: אדום (R), ירוק (G), כחול (B) וצהוב (Y). על כל קלף רשום מספר בין 1 ל-9. סה"כ משתמשים במשחק 36 קלפים שונים.

Java:

```
public class Card
{
  private char color;
  private int digit;
  public Card(char c, int d)
  {
    this.color = c;
    this.digit = d;
  }
}
```

C#:

```
public class Card
{
  private char color;
  private int digit;
  public Card(char c, int d)
  {
    this.color = c;
    this.digit = d;
  }
}
```


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

א. קלף `c1` נחשב "חזק יותר" מקלף `c2` אם המספר שרשום על הקלף `c1` גדול מהמספר שרשום על הקלף `c2`. אם שני המספרים הם זהים יש להשוות צבעים לפי סדר הבא: אדום (הכי חלש), ירוק, כחול, צהוב (הכי חזק).

כתבו במחלקה `Card` פעולה `compareTo`. הפעולה תקבל הפניה לעצם `other` מטיפוס `Card`.

- אם הקלף שמפעיל את הפעולה (`this`) "חזק יותר" מהקלף שמתקבל כפרמטר (`other`), הפעולה תחזיר -1.
- אם הקלף שמפעיל את הפעולה (`this`) "חלש יותר" מהקלף שמתקבל כפרמטר (`other`), הפעולה תחזיר 1.
- אם הקלף שמפעיל את הפעולה (`this`) זהה לקלף שמתקבל כפרמטר (`other`), הפעולה תחזיר 0.

כותרת הפעולה:

Java:

```
public int compareTo(Card other)
```

C#:

```
public int CompareTo(Card other)
```

(3 נקודות)

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

למשחק Eka כללים הבאים:

שני שחקנים לוקחים כל אחד קלף מראש החפיסה משותפת. שחקן שהקלף שלו "חזק יותר" מוסיף את זוג הקלפים לחפיסה שלו. אם אחד מהשחקנים צבר את כל הקלפים מצבע כלשהו, הוא ניצח במשחק.

אם כל הקלפים חולקו ואין מנצח - מכריזים על תיקו.

המחלקה `Deck` מייצגת חפיסת קלפים. במחלקה הוגדרו פעולות הבאות:

כותרת הפעולה (Java)	כותרת הפעולה (C#)	תיאור
`public Deck()`	`public Deck()`	פעולה בונה המייצרת חפיסה ריקה (ללא קלפים).
`public void addCard(Card c)`	`public void AddCard(Card c)`	פעולה המקבלת קלף `c` ומוסיפה אותו לחפיסה
`public Card drawCard()`	`public Card DrawCard()`	פעולה המוציאה קלף מהחפיסה ומחזירה אותו. אם החפיסה ריקה, הפעולה תחזיר `null`.
`public boolean exist(Card c)`	`public bool Exist(Card c)`	פעולה המקבלת קלף `c` ובודקת אם הוא קיים בחפיסה. אם כן - הפעולה תחזיר ערך `true`, ולא - הפעולה תחזיר ערך `false`.
`public boolean isFull(char color)`	`public bool IsFull(char color)`	פעולה המקבלת צבע `color` ובודקת אם בחפיסה יש כל הקלפים של אותו צבע. אם כן - הפעולה תחזיר ערך `true`, ולא - הפעולה תחזיר ערך `false`.

ב. כתבו פעולה חיצונית המקבלת **חפיסת קלפים מלאה** (הכוללת את כל 36 הקלפים) ומדמה את תהליך המשחק. הפעולה תדפיס הודעה על תוצאת המשחק - אחת מההודעות הבאות: "הראשון מנצח", "השני מנצח" או "תיקו".

כותרת הפעולה:

Java:

```
public static void game(Deck d)
```

C#:

```
public static void Game(Deck d)
```

(5 נקודות)

ג. כתבו את המחלקה `Deck`: הגדירו תכונה או תכונות של המחלקה וממשו את כל פעולות שהוגדרו בה.

הערה: שימו לב שבמחלקה `Card` **אין פעולות get** ואי אפשר להוסיף אותן!

(8 נקודות)

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