כללי

שקף 3:




סעיף א':

false

סעיף ב':

נמחק את הצומת 80.

סעיף ג':

נשנה את הפונקציה isBalanced כך שתחזיר את הגובה של העץ, אלא אם כן העץ אינו מאוזן, אז נחזיר false. אם עץ ריק יוחזר 0, אם עלה יוחזר 1, אחרת נחזיר את הגובה המקסימלי של תת העץ השמאלי, בתוספת 1 עבור השורש הנוכחי. לאחר מכן נחשב את ההפרש בין הגבהים של תתי העצים השמאלי והימני. אם ההפרש גדול מ-1, נחזיר false. אחרת, נחזיר את הגובה המקסימלי של תתי העצים.

الحل ١:

الخطوة ١:

First f1 = new First ("one") ← count=1 , _str= one

f1. Print() .

First: one

الخطوة ٢:

count=2

First: Tow

الخطوة ٣:

first f3 = new first (f2) ← count=3 , _str= copy Tow

f3. Print()

First: Copy Tow

الخطوة ٤:

first fy= new first() ← count=4 _str= Empty Y

.First Empty Y

الخطوة ٥:

Second s1= new second( f1, f3) ← Second , count=5 _str= copy One

s1. Print()

Second

first Copy one

first copy Tow

fgo ī:

second

first Copy Emptyy

first Taw

fgo s̄:

6

first one

first tow

second

First coly one

first coly tow

second

first coly Empty y

first tow

fgo n̄:

7

first fire

second

First coly one

first coly tow

second

first six

first six

# מבחן סוף סמסטר א' תשפ"ג

מועד ב', 20 במרץ. משך הבחינה שלוש שעות. חומר עזר מותר לשימוש: Integer.MAX_VALUE.

בשאלה זו נממש מחלקה בשם Queue שתייצג תור של מספרים שלמים. התור יתמוך בפעולות הבאות:

- `enqueue(x)` – מוסיפה את המספר x לסוף התור.
- `dequeue()` – מסירה ומחזירה את האיבר הראשון בתור.
- `isEmpty()` – מחזירה `true` אם התור ריק, ו-`false` אחרת.

**שאלה 1:**

**סעיף א:**

ממשו את המחלקה Queue בשיטת המערך המעגלי. יש להשתמש במערך בגודל קבוע 100, ולשמור את האינדקסים `head` ו-`tail` שמצביעים לתחילת ולסוף התור.

**סעיף ב:**

ממשו את הפעולה `enqueue` ברקורסיה בלבד, ללא לולאות.

**סעיף ג:**

ממשו את הפעולה `dequeue` ברקורסיה בלבד, ללא לולאות.

**שאלה 2:**

נתונה המחלקה הבאה:

```
class Integer {
    private int value;
    public Integer(int v) { value = v; }
    public int getValue() { return value; }
    public void setValue(int v) { value = v; }
}
```

נתונה גם השיטה הבאה:

```
void f(Integer x, Integer y) {
    x = y;
    y.setValue(x.getValue());
}
```

נניח כי בקטע הקוד הבא, הקריאה `f(a, b)` מתבצעת לאחר שהמשתנים `a` ו-`b` הוגדרו והוכנסו להם ערכים כלשהם:

```
Integer a = new Integer(50);
Integer b = new Integer(30);
f(a, b);
int x = 20;
```

**סעיף א:**

מה יהיה ערכו של `a.getValue()` לאחר הקריאה `f(a, b)`?

**סעיף ב:**

מה יהיה ערכו של `b.getValue()` לאחר הקריאה `f(a, b)`?

**סעיף ג:**

מה יהיה ערכו של `x` לאחר הקריאה `f(a, b)`?

**שאלה 3:**

נתונה המחלקה הבאה:

```
class Node {
    int data;
    Node next;
    Node(int d, Node n) { data = d; next = n; }
}
```

נתונה גם השיטה הבאה:

```
void g(Node p) {
    if (p.next == null) return;
    int temp = p.data;
    p.data = p.next.data;
    p.next = p.next.next;
    g(p);
    p.next = new Node(temp, p.next);
}
```

נניח כי בקטע הקוד הבא, הקריאה `g(head)` מתבצעת לאחר שהרשימה המקושרת `head` הוגדרה והוכנסו לה ערכים כלשהם:

```
Node head = new Node(10, new Node(20, new Node(30, new Node(40, new Node(50, new Node(60, new Node(70, new Node(80, null))))))));
g(head);
```

**סעיף א:**

מה יהיה ערכו של `head.data` לאחר הקריאה `g(head)`?

**סעיף ב:**

מה יהיה ערכו של `head.next.data` לאחר הקריאה `g(head)`?

**סעיף ג:**

מה יהיה ערכו של `head.next.next.data` לאחר הקריאה `g(head)`?

**סעיף ד:**

מה יהיה ערכו של `head.next.next.next.data` לאחר הקריאה `g(head)`?