

پاسخ تمرین های سری اول ساختمان داده (لینک لیست و آرایه)

1)

//check mat[1..n][1..n] is a symmetric matrix

function isSymmetric(n,mat) :

for i = 1 to n :

for j = 1 to n :

if mat[i][j]<0 or mat[i][j] != mat[n-i+1][n-j+1] :

return False

return True

2)

```
function Union(LinkedList list1, LinkedList list2) {
    Node t1 = list1.head, t2 = list2.head;
    while (t1 != null) {
        push(t1.data);
        t1 = t1.next;
    }
    while (t2 != null) {
        if (!isPresent(unionList.head, t2.data))
            push(t2.data);
        t2 = t2.next;
    }
}

function Intersection(LinkedList list1, LinkedList list2) {
    Node t1 = list1.head;
    while (t1 != null) {
        if (isPresent(list2.head, t1.data))
            push(t1.data);
        t1 = t1.next;
    }
}

function Difference(LinkedList list1, LinkedList list2) {
    Node t1 = list1.head;
    while (t1 != null) {
        if (!isPresent(list2.head, t1.data))
            push(t1.data);
        t1 = t1.next;
    }
}

function push(int new_data) {
    Node new_node = new Node(new_data);
    new_node.next = head;
    head = new_node;
}

boolean isPresent(Node head, int data) {
    Node t = head;
    while (t != null)
    {
        if (t.data == data)
            return true;
        t = t.next;
    }
    return false;
}
```

3)

چون سوال خواسته که تشخیص دهیم آیا دو لیست یکسان با شروع متفاوت هستند، یا دو لیست متفاوت هستند؛ بنا بر این حالتی که دو لیست یکسان و شروع هم یکسان را در نظر نمیگیریم و دو head را با هم مقایسه نمی کنیم.

```
void Func (node Lhead , node Mhead){

    node Ltmp = Lhead.next;
    node Mtmp = Mhead.next;

    while (Ltmp != Lhead){

        if (Ltmp == Mhead){

            Ltmp = Ltmp.next;

            while ( Ltmp == Mtmp    &&    Mtmp != Mhead ){
                Ltmp = Ltmp.next;
                Mtmp = Mtmp.next;
            }

            if (Mtmp = Mhead){
                Print("یک لیست با شروع متفاوت هستند");
                End;
            }

            else {
                Print("دو لیست متفاوت هستند");
                End;
            }

        }

        else

            Ltmp = Ltmp.next;
    }

}
```

4) java

```
1 package DS_Chapter1 ;
2 import java.util.Scanner;
3
4 public class Soal4 {
5     public static void main(String[] args) {
6         Scanner s = new Scanner(System.in);
7         BigInteger c = BigInteger.add(new BigInteger(s.next()), new BigInteger(s.next()));
8         System.out.println(c.toString());
9     }
10 }
11 class BigInteger {
12     private int[] huge = new int[33];
13     private int length;
14     public BigInteger() { }
15     public BigInteger(String s) {
16         length = s.length();
17         int i = 33;
18         int j = s.length();
19         while (j != 0) {
20             i--; j--;
21             this.huge[i] = s.charAt(j) - 48;
22         }
23     }
24
25     public static BigInteger add(BigInteger a, BigInteger b) {
26         BigInteger c = new BigInteger();
27         if (a.length > b.length)
28             c.length = a.length;
29         else
30             c.length = b.length;
31
32         for (int i = 32; i >= 0; i--) {
33             if (a.huge[i] + b.huge[i] > 9 && i != 0 && i != 33 - c.length) {
34                 c.huge[i] = a.huge[i] + b.huge[i] - 10;
35                 a.huge[i - 1]++;
36             } else
37                 c.huge[i] = a.huge[i] + b.huge[i];
38         }
```

```
35         a.huge[i - 1]++;
36     } else
37         c.huge[i] = a.huge[i] + b.huge[i];
38     }
39     return c;
40 }
41
42 public boolean isZero(){
43     for (int i = 0; i < 33; i++) {
44         if(this.huge[i] != 0)
45             return false;
46     }
47     return true;
48 }
49
50 public String toString() {
51     if (this.isZero())
52         return "0";
53     for (int i = 0; i < 33; i++) {
54         if (this.huge[i] != 0) {
55             this.length = 33 - i;
56             break;
57         }
58     }
59     String s = "";
60     for (int i = 33 - length; i < 33; i++) {
61         s += this.huge[i];
62     }
63     return s;
64 }
65 }
```

```
1 package DS_Chapter1 ;
2
3 import java.util.Scanner;
4
5 public class Soal5 {
6     static Scanner in = new Scanner(System.in);
7     public static void main(String[] args){
8         int numberOfMatrices = in.nextInt();
9         for (int i = 0; i<numberOfMatrices; i++){
10             String status = "Tame";
11             int rows = in.nextInt();
12             int columns = in.nextInt();
13             int[][] matrix = new int[rows][columns];
14             for (int r = 0; r<rows; r++){
15                 int[] row = new int[columns];
16                 for (int c = 0; c<columns; c++){
17                     row[c] = in.nextInt();
18                     matrix[r][c] = row[c];
19                 }
20                 if (!isTame(row)){
21                     status = "Wicked";
22                     break;
23                 }
24             }
25             if (status != "Wicked"){
26                 for (int c = 0; c<columns; c++){
27                     int[] column = new int[rows];
28                     for (int r = 0; r<rows; r++){
29                         column[r] = matrix[r][c];
30                     }
31                     if (!isTame(column)){
32                         status = "Wicked";
```

```

29         column[r] = matrix[r][c];
30     }
31     if (!isTame(column)){
32         status = "Wicked";
33         break;
34     }
35 }
36 System.out.println(status);
37 } else {
38     System.out.println(status);
39 }
40 }
41
42 }
43
44 @ public static boolean isTame(int... arr){
45     boolean b = true;
46     int n = arr.length;
47     int [] khosh = new int[n-1];
48     for (int i = 0; i < n-1; i++)
49         khosh[i] = Math.abs(arr[i] - arr[i+1]);
50     for (int i = 0; i < n-2; i++) {
51         if(Math.abs(khosh[i] - khosh[i+1]) != 1)
52             b = false;
53     }
54     return b;
55 }
56 }

```

6) java

```
1  package DS_Chapter1;
2
3  import java.util.Scanner;
4
5  public class Soal6
6  {
7      public static void main(String[] args)
8      {
9          Scanner sc = new Scanner(System.in);
10         int n = sc.nextInt();
11         DoublyLinkedList list = new DoublyLinkedList();
12         for (int i = 1 ; i<=n ; i++)
13         {
14             list.add(i);
15         }
16         int m = sc.nextInt();
17         for (int j =0 ; j<m ; j++)
18         {
19             String command = sc.next();
20             if(command.equals("NAME"))
21             {
22                 System.out.println(list.get(sc.nextInt()));
23             }
24             else
25             {
26                 int l = sc.nextInt() ;
27                 int h = sc.nextInt() ;
28                 list.reverse(l, h);
29             }
30         }
31     }
32 }
```



```

34     class Node
35     {
36         int data ;
37         Node next ;
38         Node prev ;
39     @   public Node (int x , Node prev , Node next)
40     {
41         this.data = x ;
42         this.next = next ;
43         this.prev = prev ;
44     }
45 }
46     class DoublyLinkedList
47     {
48         Node head ;
49         Node tail ;
50     @   public DoublyLinkedList()
51     {
52     }
53     public void add(int x)
54     {
55         Node newNode = new Node(x , tail , next: null);
56         if (head == null)
57         {
58             head = tail = newNode ;
59         }
60         else
61         {
62             tail.next = newNode ;
63             tail = newNode ;
64         }

```

```
65     }
66     @private Node getNode(int index)
67     {
68         int k = 1 ;
69         Node t = head ;
70         while(index>k)
71         {
72             if(t.next != null)
73                 t = t.next ;
74             else
75                 throw new IndexOutOfBoundsException();
76             k++ ;
77         }
78         return t ;
79
80     }
81     public int get(int index) { return getNode(index).data ; }
85     public void reverse(int l , int h)
86     {
87         Node start = getNode(l);
88         Node end = getNode(h);
89         while (l<h)
90         {
91             int temp = start.data ;
92             start.data = end.data ;
93             end.data = temp ;
94             l++ ;
95             h-- ;
96             start = start.next ;
97             end = end.prev ;
98         }
99     }
```