Java113 final

1. Write a program that uses a stack to determine whether a string is palindrome.

Answer: You can use a stack to determine if a string is palindromic by pushing each character onto the stack, and then reading the string again while popping each saved character from the stack. If they match, then the string is palindromic.

```
1
 2 import java.util.*;
 3
 4 public class ListTest {
 5
 6 public static void main( String args[] )
 7 {
 8 Scanner consol = new Scanner(System.in);
 9 stack list = new stack();  // create the List container
10
11 String str ;
12 System.out.println(" Enter a string : ");
13 str = consol.nextLine();
14
15 for(int i = 0 ; i < str.length(); i++)
16 list.push(str.charAt(i) ); //push same code to insertAtfrom front
17
18
19 String newStr = "" ;
20
21 while(! list.isEmpty() )
22 {
23 newStr += list.pop() ; // pop same code to removeFromFront
24 }
25
26 if(newStr.equals(str) )
27 System.out.println("  string is palindrom . ");
28 else
29 System.out.println("  string is not palindrom . ");
30
31
32
33  }// end main
34
35  } // end class ListTest
36
```

==========

Question 1

A –

Suppose the classes **Truck** and **Motorcycle** are derived from the class **Vehicle** . identify each declaration as valid or invalid .

1. Truck t = new Vehicle() ; invalid

2. Vehicle v = new Motorcycle(); valid
3. Motorcycle m = new Truck(); invalid

Extend the class List that we have seen in the lecture by adding the following method:

A method called " getKThElement(int) " that returns the data of the Kth element of the list without deleting the node . if no such element exists , output an appropriate message .

Ex : if our list contains the numbers : 10 , 5 , 4 , 7 , 9 , 20 then getKThElement(1) should return 10 and getKThElement(2) should return 5 and so on . getKThElement(7) should out put the message " There is no such element" . A call of getKThElement on an empty list should return the massage " The list is empty " .

Public class List {

Private ListNode firstNode ;

Private ListNode lastNode ;

Private String name ;

.----------

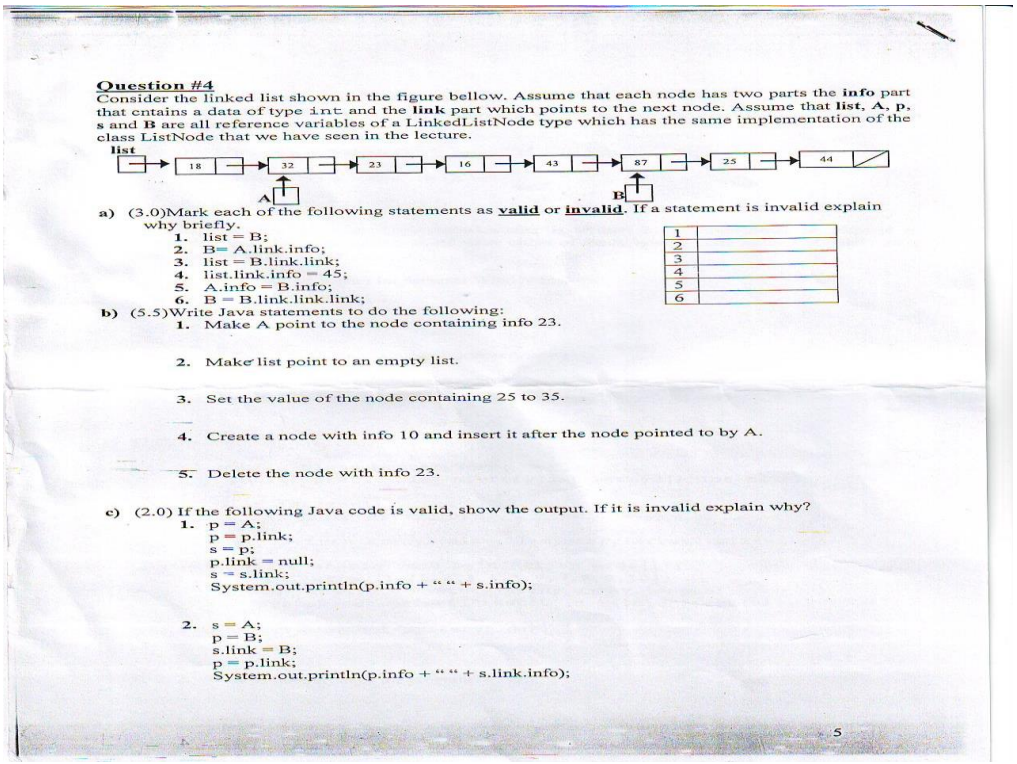
الحل:

```
public Object getKThElement(int loc)
{

if ( isEmpty() )
   {
  return  ="The list is empty";
   }


ListNode current = firstNode ;
int count = 1 ;

while(current != null)
{
if(count == loc)
return current.data ;
else {
count++;
current = current.nextNode ; }
}
   return  ="There is no such element";

}// end method getKThElement
```

نهاية الإجابة----------

} // end class List

==========



Question #4

Consider the linked list shown in the figure bellow. Assume that each node has two parts the **info** part that cntains a data of type int and the **link** part which points to the next node. Assume that **list, A, p, s** and **B** are all reference variables of a LinkedListNode type which has the same implementation of the class ListNode that we have seen in the lecture.

a) (3.0)Mark each of the following statements as **valid** or **invalid**. If a statement is invalid explain why briefly.

1. list = B;
2. B= A.link.info;
3. list = B.link.link;
4. list.link.info = 45;
5. A.info = B.info;
6. B = B.link.link.link;

1	
2	
3	
4	
5	
6	

b) (5.5)Write Java statements to do the following:

1. Make A point to the node containing info 23.
2. Make list point to an empty list.
3. Set the value of the node containing 25 to 35.
4. Create a node with info 10 and insert it after the node pointed to by A.
5. Delete the node with info 23.

c) (2.0) If the following Java code is valid, show the output. If it is invalid explain why?

1.
```
p = A;
p = p.link;
s = p;
p.link = null;
s = s.link;
System.out.println(p.info + " " + s.info);
```

2.
```
s = A;
p = B;
s.link = B;
p = p.link;
System.out.println(p.info + " " + s.link.info);
```

5

Question 4

السؤال موجود في الصورة المرفقة ايضا مع الوورد

ملاحظة مهمة : الحل على اساس ان اول نود اسمها List وليس كامل الشكل كائن من نوع list

الظاهرة في الصورة هو كل السلسلة فإن كثير من الحلول سوف تختلف تماما . listوفي حال كان المقصود بال

الحل

A)

1. true .

2. false . عدم توافق في الأنواع.

3. true .

4. true . على اعتبار ان الجملة منفصلة عن السابق و الا فهي خاطئة

5. true .

6. true . لكن هذا صحيح nullسسوف تشير لكائن

b)

1. Made A point to the node containing info 23 .

A = A.link .

2. make list point to an empty list .

list الموجودة و الأن المفروض تشير ل node أحد ال listهنا السؤال مش واضح لأن في البداية قالو في السؤال ان كاملة لكنها فارغة

```
List   mylist = new List () ;

list = mylist . firstnNode ;
```

فالحل يكون كالتالي listأما اذا كان المفورض انها

```
List   mylist = new List () ;

list = mylist  ;
```

3. set the value of the node containing 25 to 35 .

```
B.link.info = 35  ;
```

4. create a node with info 10 and insert it after the node pointed to by A .

```
A.link = new LinkedListNode  (10 , A.link ) ;
```

5. delete the node with info 23 .

```
A.link = A.link.link ;
```

C)

1. في البداية سوف يطبع 23 ثم يظهر رسالة خطأ لذلك :

This code false because line 5 (s = s.link) point to null node .

2.

true code . the out put :

25 87
// لاحظو في السطر الثالث من الكود قم اقتطاع الكثير من النود بسبب الجملة التالية

// s.link = B ;

Fill the array (array) in list

```
//insert in list
30 for(int i=0 ; i<25 ; i++)
31 list.insertAtFront(array[i]);
32
```

==========

Copy list1 in list2 in revers order

```
Geven :      list2.revers_copy(list);
```

```
public void revers_copy(List list1)
 {
 while(list1.firstNode != null)
 {
 insertAtBack(list1.removeFromBack());
 }//end while

 }// end method
```

==========

Delete small element in the list

```
1 public void deleteSmallest()
 2 {
```

```
 3  if ( isEmpty() ) {
 4    System.out.println( "Empty list");
 5    return;
 6    }
 7
 8    ListNode current = firstNode;
 9    int small =(Integer) current.data ;
10
11
12     while ( current != null )
13     {
14   if( (Integer)current.data < small )
15   small = (Integer)current.data ;
16
17     current = current.nextNode;
18     }
19     //  now delet the small data
20
21       if((Integer)firstNode.data == small)
22          removeFromFront();
23 else
24 {
25    ListNode priv = firstNode;
26      current = firstNode.nextNode ;
27
28  while ( current != null )
29 {
30         if((Integer) current.data == small )
31         {
32          priv.nextNode = current.nextNode ;
33          return ;
34         }
35     priv = priv.nextNode ;
36     current = current.nextNode;
37 }
38 }// else
39
40 }
41
```

==========

أكواد الإضافة و الحذف بدون ما يكون عندي

last

او بدون ما يكون عندي

tail

```
public void insertAtBack(object item)
{
if( first == null )
{
System.out.println("empty list") ;
return ;
}
```

```
Node current = first ;

while( current.next != null )
current = current .next ;

current .next = new Node(item)



}

=======================================
public object removeFromBack()
{
if( first == null )
{
System.out.println("empty list") ;
return null ;
}

Node current = first ;

while( current.next.next != null )
current = current .next ;

object = current.next.data ; // الإحتفاظ بالبيانات المراد حذفها
current .next = null ;

return object ;


}
```

بعد صلآتي من قال (بسم الله الرحمن الرحيم ولا حول ولآ قوة إلا بالله العلي العظيم سبعا' الصبح والمغرب كتب من السعداء ولو كان من الأشقياء

مِن قرأ آلثلآث آلآيآت آلآخيره مِن سورة آلحشر في آلنهآر صلى عليه 70 آلف مِلك حتى يمِسي و آذآ قرأهآ آلمِسآء صلى عليه 70 آلف مِلك حتى يصبح و آذآ مِآت مِآت شهي