Solutions de l'Examen du cours de Programmation et Algorithmique II
1ère Session, Mai 2016

NOM : ..... PRENOM : ..... SECTION : .....

Partie 1 – Question 1

```
1  public static boolean isFirst(char c) {
2    return (c == '_') || (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z');
3  }
4
5  public static boolean isNext(char c) {
6    return isFirst(c) || (c >= '0' && c <= '9');
7  }
8
9  public static void checkIdentifier(String s)
10   throws InvalidIdentifierException {
11   if (s.length() == 0)
12     throw new InvalidIdentifierException("an␣empty␣string␣is␣not␣a␣valid␣identifier");
13   if (!isFirst(s.charAt(0)))
14     throw new InvalidIdentifierException(s.charAt(0) + "␣cannot␣be␣used␣as␣first␣character");
15   for (int i= 1; i < s.length(); i++)
16     if (!isNext(s.charAt(i)))
17       throw new InvalidIdentifierException(s.charAt(i) + "␣cannot␣be␣used␣in␣an␣identifier");
18 }
19
20 // For testing purposes : was not required in exam
21 public static void testCheckIdentifier(String s) {
22   System.out.print("\"" + s + "\"␣->␣");
23   try {
24     checkIdentifier(s);
25     System.out.println("␣OK");
26   } catch (InvalidIdentifierException e) {
27     System.out.println(e.getMessage());
28   }
29 }
30
31 // For testing purposes : was not required in exam
32 public static void main(String [] args) {
33   testCheckIdentifier("234");
34   testCheckIdentifier("boule8ç");
35   testCheckIdentifier("_34");
36   testCheckIdentifier("trululu");
37   testCheckIdentifier("");
38   testCheckIdentifier("x");
39 }
```

Solutions de l'Examen du cours de Programmation et Algorithmique II
1ère Session, Mai 2016

NOM : ..... PRENOM : ..... SECTION : .....

Partie 1 – Question 2

```
1  public static String longestCommonSubstring(String a, String b) {
2    // maintain (length, position) of longest substring so far
3    int maxLen= 0;
4    int maxEnd= -1;
5    // memoization : maintain already computed values of L(m,n)
6    int [][] t= new int [a.length()][b.length()];
7
8    for (int i= 0; i < a.length(); i++) {
9      for (int j= 0; j < b.length(); j++) {
10       if (a.charAt(i) == b.charAt(j)) {
11         if ((i == 0) || (j == 0)) {
12           t[i][j]= 1;
13         } else {
14           t[i][j]= t[i-1][j-1] + 1;
15         }
16         if (t[i][j] > maxLen) {
17           maxLen= t[i][j];
18           maxEnd= i;
19         }
20       } else {
21         t[i][j]= 0;
22       }
23     }
24   }
25
26   if (maxLen > 0)
27     return a.substring(maxEnd-maxLen+1, maxEnd+1);
28   return "";
29 }
```

Solutions de l'Examen du cours de Programmation et Algorithmique II
1ère Session, Mai 2016

NOM : ..... PRENOM : ..... SECTION : .....

Partie 2 – Question 1

```
1  private class _Iterator implements Iterator<E> {
2
3    private Node next= head;
4
5    public boolean hasNext() {
6      return (next != null);
7    }
8
9    public E next() {
10     Node tmp= next;
11     next= tmp.next;
12     return tmp.e;
13   }
14
15 }
```

```
1  public boolean add(E e) {
2    if (contains(e))
3      return false;
4    Node tmp= head;
5    head= new Node();
6    head.e= e;
7    head.next= tmp;
8    size++;
9    return true;
10 }
```

```
1  public boolean addAll(Collection<? extends E> c) {
2    boolean ok= false;
3    for (E e: c)
4      ok|= add(e);
5    return ok;
6  }
```

```
1  public void clear() {
2    head= null;
3    size= 0;
4  }
```

```
1  // support method used in 'contains' and 'remove'
2  private static <E> boolean _equals(E a, E b) {
3    return ((((a == null) && (b == null)) ||
4             ((a != null) && a.equals(b))));
5  }
```

```
1  public boolean contains(Object o) {
2    Node tmp= head;
3    while (tmp != null) {
4      if (_equals(tmp.e, o))
5        return true;
6      tmp= tmp.next;
7    }
8    return false;
9  }
```

Solutions de l'Examen du cours de Programmation et Algorithmique II

1ère Session, Mai 2016

NOM : ..... PRENOM : ..... SECTION : .....

```java
public Iterator<E> iterator() {
  return new _Iterator();
}
```

```java
public boolean remove(Object o) {
  if (head == null)
    return false;
  if (_equals(head.e, o)) {
    head= head.next;
    size--;
    return true;
  }
  Node prev= head;
  while (prev.next != null) {
    if (_equals(prev.next.e, o)) {
      prev.next= prev.next.next;
      size--;
      return true;
    }
    prev= prev.next;
  }
  return false;
}
```

Solutions de l'Examen du cours de Programmation et Algorithmique II
1ère Session, Mai 2016

NOM : ..... PRENOM : ..... SECTION : .....

Partie 2 – Question 2

Cette question peut être résolue en compilant et exécutant le programme fourni dans l'énoncé.