University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2009

Practical Test 1A

Time: 45 minutes

Leetspeak

Write a program to replace every occurrence of the character "e" with the number "3" in a string that is provided as input. The method that does the conversion MUST use recursion.

Use the Scanner class for input. You may not use loops in any part of the program!

Sample I/O:

```
Hello
H3llo
```

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1A
{
   public static void main ( String [] args )
   { (new Test_Leet ()).run(); }
}

class Test_Leet
{
   void run ()
   {
      Scanner input = new Scanner (System.in);
      String text = input.nextLine ();
      System.out.println (leet (text));
   }

   public String leet ( String text )
   {
   // write your code here
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the **Test1A.java** source file contained within a .ZIP file to the Automatic Marker. Marks will be awarded for correctness only – in addition, the use of any loop construct in your code will result in a penalty of 60%!

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2009

Practical Test 1B

Time: 45 minutes

Repetitions

Write a program to check if a given string is composed of entirely the same character or not, and return a boolean result. The method that does the checking MUST use recursion.

Use the Scanner class for input. You may not use loops in any part of the program!

Sample I/O:

```
HHHHH
true
```

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1B
{
   public static void main ( String [] args )
   { (new Test_Repeat ()).run(); }
}

class Test_Repeat
{
   void run ()
   {
      Scanner input = new Scanner (System.in);
      String text = input.nextLine ();
      System.out.println (isRepeat (text));
   }

   public boolean isRepeat ( String text )
   {
   // write your code here
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the **Test1B.java** source file contained within a .ZIP file to the Automatic Marker. Marks will be awarded for correctness only – in addition, the use of any loop construct in your code will result in a penalty of 60%!

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2009

Practical Test 1C

Time: 45 minutes

DNA Checker

Write a program to check if a given string is a DNA sequence (sequence of A, T, G or C characters) or not, and return a boolean result. The method that does the checking MUST use recursion.

Use the Scanner class for input. You may not use loops in any part of the program!

Sample I/O:

```
ATBDCG
false
```

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1C
{
   public static void main ( String [] args )
   { (new Test_DNA ()).run(); }
}

class Test_DNA
{
   void run ()
   {
      Scanner input = new Scanner (System.in);
      String text = input.nextLine ();
      System.out.println (isDNA (text));
   }

   public boolean isDNA ( String text )
   {
   // write your code here
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the **Test1C.java** source file contained within a .ZIP file to the Automatic Marker. Marks will be awarded for correctness only – in addition, the use of any loop construct in your code will result in a penalty of 60%!

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2009

Practical Test 1D

Time: 45 minutes

Partial Sum

Write a program to calculate the sum of positive integers from a given lower bound to a given upper bound. The method that does the calculation MUST use recursion.

Use the Scanner class for input. You may not use loops in any part of the program!

Sample I/O:

```
3
7
25
```

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1D
{
   public static void main ( String [] args )
   { (new Test_Sum ()).run(); }
}

class Test_Sum
{
   void run ()
   {
      Scanner input = new Scanner (System.in);
      int i = input.nextInt();
      int j = input.nextInt();
      System.out.println (sum (i, j));
   }

   public int sum ( int a, int b )
   {
   // write your code here
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the **Test1D.java** source file contained within a .ZIP file to the Automatic Marker. Marks will be awarded for correctness only – in addition, the use of any loop construct in your code will result in a penalty of 60%!

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2009

Practical Test 1E

Time: 45 minutes

Sequential Digits

Write a program to generate an integer containing a list of sequential digits from 1 to a given length (up to 9). The method that does the generation MUST use recursion.

Use the Scanner class for input. You may not use loops in any part of the program!

Sample I/O:

```
4
1234
```

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1E
{
   public static void main ( String [] args )
   { (new Test_Sequence ()).run(); }
}

class Test_Sequence
{
   void run ()
   {
      Scanner input = new Scanner (System.in);
      int i = input.nextInt();
      System.out.println (sequence (i));
   }

   public int sequence ( int a )
   {
   // write your code here
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the **Test1E.java** source file contained within a .ZIP file to the Automatic Marker. Marks will be awarded for correctness only – in addition, the use of any loop construct in your code will result in a penalty of 60%!

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2009

Practical Test 1F

Time: 45 minutes

Mirrors

Write a program to append the mirror image of a string to itself. The method that does the transformation MUST use recursion.

Use the Scanner class for input. You may not use loops in any part of the program!

Sample I/O:

```
Hello
HelloolleH
```

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1F
{
   public static void main ( String [] args )
   { (new Test_Mirror ()).run(); }
}

class Test_Mirror
{
   void run ()
   {
      Scanner input = new Scanner (System.in);
      String text = input.nextLine ();
      System.out.println (mirror (text));
   }

   public String mirror ( String text )
   {
   // write your code here
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the **Test1F.java** source file contained within a .ZIP file to the Automatic Marker. Marks will be awarded for correctness only – in addition, the use of any loop construct in your code will result in a penalty of 60%!

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2009

Practical Test 1G

Time: 45 minutes

Capital Counter

Write a program to count the number of capital letters in a given string. The method that does the counting MUST use recursion.

Use the Scanner class for input. You may not use loops in any part of the program!

Sample I/O:

```
Hello World
2
```

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1G
{
   public static void main ( String [] args )
   { (new Test_Caps ()).run(); }
}

class Test_Caps
{
   void run ()
   {
      Scanner input = new Scanner (System.in);
      String text = input.nextLine ();
      System.out.println (countCaps (text));
   }

   public int countCaps ( String text )
   {
   // write your code here
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the **Test1G.java** source file contained within a .ZIP file to the Automatic Marker. Marks will be awarded for correctness only – in addition, the use of any loop construct in your code will result in a penalty of 60%!

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2009

Practical Test 1H

Time: 45 minutes

Odd Sum

Write a program to find the sum of all positive odd integers less than a given maximum. The method that does the calculation MUST use recursion.

Use the Scanner class for input. You may not use loops in any part of the program!

Sample I/O:

```
10
25
```

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1H
{
   public static void main ( String [] args )
   { (new Test_Odd ()).run(); }
}

class Test_Odd
{
   void run ()
   {
      Scanner input = new Scanner (System.in);
      int i = input.nextInt();
      System.out.println (oddsum (i));
   }

   public int oddsum ( int a )
   {
   // write your code here
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the **Test1H.java** source file contained within a .ZIP file to the Automatic Marker. Marks will be awarded for correctness only – in addition, the use of any loop construct in your code will result in a penalty of 60%!

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2009

Practical Test 1J

Time: 45 minutes

Encryption

Write a program to convert every character in a given string to the next sequential Unicode character. The method that does the conversion MUST use recursion.

Use the Scanner class for input. You may not use loops in any part of the program!

Sample I/O:

```
Hello
Ifmmp
```

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1J
{
   public static void main ( String [] args )
   { (new Test_Encrypt ()).run(); }
}

class Test_Encrypt
{
   void run ()
   {
      Scanner input = new Scanner (System.in);
      String text = input.nextLine ();
      System.out.println (encrypt (text));
   }

   public String encrypt ( String text )
   {
   // write your code here
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the **Test1J.java** source file contained within a .ZIP file to the Automatic Marker. Marks will be awarded for correctness only – in addition, the use of any loop construct in your code will result in a penalty of 60%!

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2009

Practical Test 1K

Time: 45 minutes

Spammer

Write a program to insert an underscore character (_) after every character in a given string. The method that does the conversion MUST use recursion.

Use the Scanner class for input. You may not use loops in any part of the program!

Sample I/O:

```
Hello
H_e_l_l_o_
```

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1K
{
   public static void main ( String [] args )
   {
      (new Test_Insert ()).run();
   }
}

class Test_Insert
{
   void run ()
   {
      Scanner input = new Scanner (System.in);
      String text = input.nextLine ();
      System.out.println (insert (text));
   }
   public String insert ( String text )
   {
   // write your code here
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the **Test1K.java** source file contained within a .ZIP file to the Automatic Marker. Marks will be awarded for correctness only – in addition, the use of any loop construct in your code will result in a penalty of 60%!

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2009

Practical Test 1M

Time: 45 minutes

Spam Checker

Write a program to check if a string contains underscores (_) as every second character, or not, and returns **true** or **false** appropriately. The method that does the checking MUST use recursion.

Assume all strings have an even number of characters. Use the Scanner class for input. You may not use loops in any part of the program!

Sample I/O:

```
H_e_l_l_o_
true
```

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1M
{
   public static void main ( String [] args )
   {
      (new Test_Spam ()).run();
   }
}
class Test_Spam
{
   void run ()
   {
      Scanner input = new Scanner (System.in);
      String text = input.nextLine ();
      System.out.println (checkspam (text));
   }
   public boolean checkspam ( String text )
   {
   // write your code here
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the **Test1M.java** source file contained within a .ZIP file to the Automatic Marker. Marks will be awarded for correctness only – in addition, the use of any loop construct in your code will result in a penalty of 60%!