NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
POTCHEFSTROOMKAMPUS

Benodigdhede vir hierdie vraestel/Requirements for this paper:			
Multikeusekaarte/ Multi-choice cards:		Nie-programmeerbare sakrekenaar/ Non-programmable calculator:	
Grafiekpapier/ Graph paper:		Skootrekenaar/ Laptop:	

Word ander hulpmiddels toegelaat/ Are other resources allowed?
NEE/ NO

EKSAMEN/TOETS EXAMINATION/TEST:	**EERSTE Eks. 2016 FIRST EXAM 2016**	KWALIFIKASIE/ QUALIFICATION:	**BSc**
MODULEKODE/ MODULE CODE:	**ITRW222**	TYDSDUUR/ DURATION:	**3 hours**
MODULEBESKRYWING/ MODULE DESCRIPTION:	**Datastrukture/ Data Structures**	MAKS/ MAX:	**100**
EKSAMINATOR(E)/ EXAMINER(S):	**Prof. R Goede**	DATUM/ DATE:	**24/10/2016**
MODERATOR/ MODERATOR:	**Mnr. J Prinsloo**	TYD/TIME:	**14:00**

Student name: ..... University number: .....

Examination Instructions

1. Students are not allowed to handle cell phones in the examination room and cell phone accessories including but not limited to earpieces, are not allowed.
2. Students bring bags to the venue at own risk, and must put them in front of the room.
3. Students may not wear caps / hats / beanies in the examination venue.
4. Students are subject to disciplinary procedures should they:
   4.1 have books or notes in their possession (except during open book examinations);
   4.2 attempt to assist another student, or attempt to obtain assistance.
5. Students are allowed into venue in the first half hour of session, but no extra time is granted.
6. No student is allowed to leave the examination venue before half an hour of the examination session has elapsed.
7. No refreshments are allowed in the examination venue.
8. No pages may be removed from the answer scripts.
9. Before students leave the examination venue, answer scripts must be handed to the invigilators.
10. The attendance slip on the back cover that also serves as an under-taking, must be completed.
11. All examination answers must be written in black or blue ink.

Eksamenvoorskrifte

1. Studente mag nie selfone in die eksamenlokaal hanteer nie en selfoontoebehore wat insluit maar nie tot oorfone beperk is nie, is nie toelaatbaar nie.
2. Studente bring sakke na lokaal op eie risiko, en moet dit voor in die lokaal neersit.
3. Studente mag nie pette / hoede / musse in die eksamenlokaal dra nie.
4. Studente stel hulle aan dissiplinêre optrede bloot indien hulle:
   4.1 enige boeke of notas by hulle sou hê (behalwe by oopboek-eksamens);
   4.2 'n ander student probeer help of probeer om hulp te kry.
5. Studente mag in eerste halfuur van sessie tot lokaal toegelaat word, maar geen ekstra tyd word toegestaan nie.
6. Geen student word toegelaat om die eksamenlokaal te verlaat binne die eerste halfuur van 'n eksamensessie nie.
7. Geen verversings word in 'n eksamenlokaal toegelaat nie.
8. Geen bladsye mag uit die antwoordskrif verwyder word nie.
9. Voordat studente die eksamenlokaal verlaat, moet die antwoordskrifte aan die toesighouers oorhandig word.
10. Die presensiestrokie op die agterblad wat ook as onderneming geld, moet voltooi word.
11. Studente moet slegs met swart of blou penne skryf.

1480

Vraag 1/ *Question 1* (20)

1.1 Gebruik die gedetailleerde model (t-notasie) om die looptyd van die volgende programlyne te bepaal: (8)	*1.1 Use the detailed model (t-notation) to determine* running time *of the following program lines:* (8)
1. for (int i=1; i<n; i++) 2. b=2*arr[i]; (binne die lus / *inside the loop*)	
1a)	**1b)**
1c)	**2)**

1.2 Bepaal die looptyd van die ***aangeduide lyne*** in konteks van hierdie programdeel. **Maak gebruik van die vereenvoudigde model EN asimptotiese ontleding.** Jy hoef nie die uitdrukkings te vereenvoudig nie. **Kyk versigtig na die lynnommers**! (6)	*1.2 Determine the running time of the* ***specified lines*** *in context of this program segment.* ***Use the simplified model and asymptotic analysis.*** *You need not simplify the expressions.* **Carefully check the line numbers!!** (6)

```
1     public class Question1_2
2     {
3       public static int numbers (int n)
4       {
5       int ans = 1;
6       for (int i=1;  i<n;  i++ )
7        {
8           for ( int j=0;  j<=i; ++j)
9             ans =(ans*i)+2;
10       }
11      return prod;
12    }
13    }
```

	VEREENVOUDIGDE MODEL / *SIMPLIFIED MODEL*	**ASIMPTOTIESE MODEL / *ASYMPTOTIC MODEL***
6a		
8b		
9		

1481

1.3 Bewys die volgende gelykheid: (4)	*1.3 Proof the following equation:* (4)
$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$	

1.4 Gee die definisie van die asimptotiese bogrens – groot O. (2)	*1.4 Give the definition for the asymptotic upper bound – big Oh.* (2)

Vraag 2 / *Question 2* (30)

Bestudeer die volgende klas:	*Study the following class:*

```
public class SLLNode<T extends Comparable<T>> {
    public T info;
    public SLLNode<T> next;
    public SLLNode() {
        this(null,null);
    }
    public SLLNode(T el) {
        this(el,null);
    }
    public SLLNode(T el, SLLNode<T> ptr) {
        info = el; next = ptr;
    }
}

public class SLL<T extends Comparable<T>>
{
    protected SLLNode<T> head, tail;
    public SLL() {
        head = tail = null;
    }
    public boolean isEmpty() {
        return head == null;
    }

  public void printAll() {
          //  code that displays all elements
```

```
            //  kode wat al die elemente vertoon
    }
}
```

2.1 Skryf 'n metode in Java vir die klas wat 'n element agter aan die lys sal voeg. (5)	*2.1 Write a method in Java for the class that will add an element to the back of the list. (5)*

2.2 Ontwerp 'n metode genaamd: ***int ListStats()*** wat in leë geskakelde lys as parameter ontvang en in sy plek 'n lys wat uit 2 elemente bestaan sal skep. Hierdie nuwe parameterlys moet bestaan uit die kleinste en die grootste elemente van die roepende lys. Die metode moet 'n getal terugstuur wat gelyk is aan die aantal elemente in die lys. Jy moet eers die parameterlys skoonmaak om seker te maak dit is leeg. *Roepende lys = {3,1,5,8,6,5,4,3,7,1,8}* *Parameter lys = {} //ontvang lee lys* *NA int listStatst(….)* *Roepende lys = onveranderd* *Parameterlys = {1,8}* *Terugstuurwaarde= 11* Jy moet die kode gee vir al die bestaande SLL metodes wat jy wil gebruik, behalwe printAll().	*2.2 Design a method for the class called:* **int ListStats(…)** *that receives an empty linked list as parameter and should create in its place a list consisting of two elements. This new parameter list should consist of the smallest and largest elements of the calling list. The method should return a number equal to the number of elements in the list. You must first clean the parameter list to insure that it is empty.* *Calling List= {3,1,5,8,6,5,4,3,7,1,8}* *Parameter list ={l} // receives empty list* *AFTER int listStats(…)* *Calling list = unchanged* *Parameter list= {1,8}* *Return value = 11* You have to give all the code for existing SLL methods you want to use, except printAll().
2.2.1 Teken 'n geskakelde lys om jou met die algoritme-ontwerp te help. (2)	*2.2.1 Draw a linked list to help you to design the algorithm.* (2)

2.2.2 Skryf die algemene en die spesiale gevalle vir die probleem in Afrikaans neer. Gee 'n kort beskrywing van die **nodige aksie vir** elkeen van die gevalle. (2)	*2.2.2 Write down the general and all the special cases for the problem in English. Give a short description of the **required action for** each of these cases. (2)*
2.2.3 Skryf die metode: ***int listStats(....)*** in Java. (13)	*2.2.3 Write the method: **int listStats(....)** in Java.* (13)

2.2.4 Skryf 'n drywer program om die metode deeglik te toets. (8)	2.2.4. *Write a driver program to test the method thoroughly. (8)*

Vraag 3 / *Question 3 (15)*

3.1 Verduidelik hoe 'n stapel in Java gebruik kan word om te verseker dat hakies ({}[]()) korrek gebruik word. (5)	*3.1 Explain how a stack can be used in Java to ensure that brackets ({}[]()) are used correctly.* (5)

3.2 As jy 'n tou het, wat geimplementer is met behulp van 'n sikliese skikking met veranderlikes: first en last wat die eerste en laaste elemente van die gevulde tou aandui, skryf die metode **enqueue()** om 'n element in die tou te voeg.
Gebruik die volgende kode as basis: (10)

*3.2 If you have a queue implemented with a circular array with variables: first and last indicating the first and last element of the used section of the queue –write the method **enqueue()** to add an element to the queue.*
Use the following code as base: (10)

```
public class ArrayQueue {
    private int first, last, size;
        int count =0;
    private Object[] storage;
    public ArrayQueue() {
        this(100);
    }
    public ArrayQueue(int n) {
        size = n;
        count = 0;
        storage = new Object[size];
        first = 0;
        last = size-1;
    }
```

Vraag 4 / *Question 4 (10)*

4.1 Verduidelik elkeen van die volgende terme in jou eie woorde – maak seker jy elkeen onderskei van die ander wat gevra word: (5)

4,1 Explain each of the following terms in your own words – take care to distinguish each one from the others specified : (5)

Rekursie/ recursion:

Stert rekursie/ tail recursion:

1486

Nie-stert rekursie/ nontail recursion:

Indirekte rekursie/ indirect recursion:

Geneste rekursie / nested recursion:

4,2 Bereken die faktoriaal van *n*. Dit word wiskundig bereken as *n**(*n*-1)*(*n*-2)*....*1.

Die faktoriaal van 1 =1 = 1
Die faktoriaal van 2 =2= (2*1)
Die faktoriaal van 3 = 6 = (3*2*1)

Jy moet die faktoriaal van ***n*** rekursief (sonder lusse) bereken.

Voltooi die kode: (5)

4.2 Compute the factorial of n. *It is computed mathematically as* n*(n-1)*(n-2)*....*1.

The factorial van 1 =1= (1)
*The factorial van 2 =2 =(2*1)*
*The factorial van 3 = 6 = (3*2*1)*

You have to compute the factorial of **n** *recursively (without loops).*

Complete the code: *(5)*

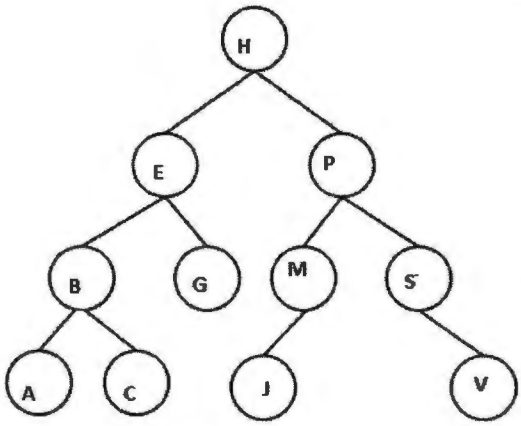
```
public int factorial(int n) {




}
```

1487

Vraag 5 / Question 5 (10)



5.1 Gee die volgorde waarmee die volgende algoritmes deur die boom sal beweeg: (4)	*5.1 Give the order in which the following algorithms will traverse the tree: (4)*

Breadth-first:

Pre-order:

Blaai asb om / *Please turn over*

5.2 Beskou die volgende kode. **Voltooi** die gegewe kode van die metode: *public T insert(T el)* **wat die parameter element in die boom sal voeg.** **(5)**	5.2 Consider the following code. **Complete** the code for the method: ***public T insert(T el)*** *which will insert the parameter element in the tree).* ***(5)***

```
public class BSTNode<T extends Comparable<? super T>> {
   protected T el;
   protected BSTNode<T> left, right;
   public BSTNode() {
      left = right = null;
   }
   public BSTNode(T el) {
      this(el,null,null);
   }
   public BSTNode(T el, BSTNode<T> lt, BSTNode<T> rt) {
      this.el = el; left = lt; right = rt;
   }
}
public class BST<T extends Comparable<? super T>> {
   protected BSTNode<T> root = null;
   public BST() {
   }
}
```

Voltooi *die metode: insert()*	***Complete*** *the method: insert()*

```
public void insert(T el) {
      BSTNode<T> p = root, prev = null;
      while (p != null) {  // find a place for inserting new node;
         prev = p;
         if (el.compareTo(p.el) < 0)
             p = p.left;
         else p = p.right;
      } //now add new element……

      // add your code here









}
```

Vraag 6 / *Question 6 (15)*

6.1 **Bespreek** en dui op 'n diagram aan hoe die **Bucketsort** algoritme die volgende getalle sal sorteer: (7)	**6.1** ***Discuss*** *and indicate by means of a diagram how the* **Bucketsort** *algorithm will sort the following list of numbers:* *(7)*

1 4 1 3 1 4 2 3 2 4

6.2 Voltooi die Java kode van **quicksort:** (8)	*6.2 Complete the Java code for **quicksort:*** *(8)*

```
1   void quicksort(T[] data) {
2           if (data.length < 2)
3                   return;
4           int max = 0;
5           // find the largest element and put it at the end of data:
6           for (int i = 1; i < data.length; i++)
7                   if (data[max].compareTo(data[i]) < 0)
8                           max = i;
9           swap(data,data.length-1,max); // largest el is now in its
10          quicksort(data,0,data.length-2); // final position;
11  }
12
13  <T extends Comparable<? super T>> void quicksort(T[] data, int first, int last) {
14          int lower = first + 1, upper = last;
15          swap(data,first,(first+last)/2);
16          T bound = data[first];
            //ADD YOUR CODE HERE
```