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

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:	**FIRST EXAM 2017**	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:	30-10-2017
MODERATOR/ MODERATOR:	Mnr. J Prinsloo	TYD/TIME:	**14:00**

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

Examination Instructions

1. Students are not allowed to handle cell phones/ smart watches 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 / slimhorlosies 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.

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=arr[i]+submethod(i);   // binne die lus en  submethod () is ‘n metode
    }                              //inside the loop and   submethod()  is a method
```

1a)	**1b)**
1c)	**2)**

1.2 Bepaal die looptyd van die ***aangeduide lyne*** <u>in konteks van hierdie programdeel</u>. **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* ***<u>specified lines</u>*** *<u>in context of this program segment</u>.* ***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=0;  i<=n;  i++ )
7            {
8               for ( int j=1;  j<=i+1; ++j)
9                  ans =ans+i+2;
10           }
11         return prod;
12       }
13      }
```

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

1.3 Bewys die volgende gelykheid: (4)	*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 SLL<T extends Comparable<? super T>>
{
   private Element<T> head;  // list header
   private Element<T> tail;

    public SLL()
    {  head = null;
       tail = null;
    }
public class Element<T extends Comparable<? super T>>
  {
      private T data;
      private Element<T> next;
      public Element(T param)
     {
             data = param;
     }
   }// end of inner class Node
}//end SLL outer class
    }
}
```

2.1 Skryf 'n metode ***prepend()*** in Java vir die klas wat element voor in die lys sal byvoeg. Jy hoef nie die beplanning te wys nie. (5)	*2.1 Write a method* **prepend** *in Java for the class that will add an element in front of the list. You need not show the planning. (5)*

2.2 Ontwerp 'n metode genaamd: ***deletePos(...)*** om 'n element volgens positisie uit 'n geskakelde lys te verwyder. Jou metodoe ontvang 'n ENKELWAARDE as parameter. Aanvaar die eerste element in die lys is by posisie 0. *Roepende lys = {3,1,5,8,6,5}* *Parameter waarde = 2* *NA deletePos():* *Roepende lys = {3,1,8,6,5} en 'n 5 is teruggestuur.* Jy moet die kode gee vir al die bestaande SLL metodes wat jy wil gebruik, **behalwe** toStringl(). Jy mag hierdie metode gebruik sonder om die kode te gee.	*2.2 Design a method for the class called:* **deletePos(...)** *to remove a certain element according to its position from the list. Your method receives a SINGLE VALUE as parameter. Assume the first element of the list is at position 0.* *Calling List= {3,1,5,8,6,5}* *Parameter Value = 2* *AFTER deletePost():* *Calling List = {3,1,8,6,5 and a 5 is returned.* *You have to give all the code for existing SLL methods you want to use,* ***except*** *toString()). You may use the method without supplying the code.*
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: **deletePos(...)** in Java. (13)	*2.2.3 Write the method:* ***deletePos*** *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 gebruik kan word om baie groot getalle by mekaar te tel – jy kan jou verduideliking ondersteun met 'n diagram. (5)	*3.1 Explain how a stack can be used in Java to add very large numbers – you may complement your explanation with a diagram (5)*
3.2 As jy 'n tou 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 **dequeue()** om 'n element uit die tou te verwyder. 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 **dequeue()** to delete an element from 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 (8)*

4.1 Verduidelik in jou eie woorde hoe 'n "hash" tabel data stoor. (3)	4.1 Explain in your own words how a hash table stores data. (3)

4.2 Skryf 'n metode in java wat x^n rekursief bereken. Voltooi die kode: (7)	*4.2 Write a method that computes x^n recursively.* *Complete the code:* *(7)*

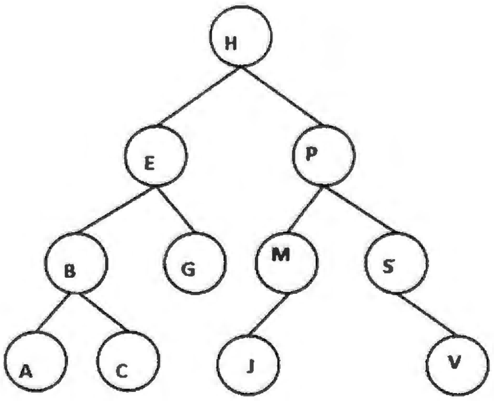
```
public double power(double x, int n) {






}
```

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)*

Pre-Order:

In-order:

5.2 Gee die pseudokode om 'n waarde in 'n binêre boom te voeg. (6)	*5.2 Give the pseudo code to insert a value in a binary tree. (6*

Vraag 6 / *Question 6* (15)

6.1 **Bespreek** en dui op 'n diagram aan hoe die **Bucketsort** algoritme die volgende getalle sal sorteer: (10)	*6.1 **Discuss** and indicate by means of a diagram how the **Bucketsort** algorithm will sort the following list of numbers: (10)*

4 4 1 2 3 2 3 1 1 2 2 3

6.2 Gee die Java kode vir seleksiesortering. (5)	*6.2 Give the Java code for selection sort. (5)*

```
public void selectionSort (T [] data)
{
    int i, j, least;





}
```