A black text on a white background

Description automatically generated

**Data Structures & Algorithms (DSA)**

Year 2, Semester 4

## SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in Information Technology

Diploma in Cybersecurity & Digital Forensics

**COMMON TEST Sample paper**

Weightage: 20%

INSTRUCTIONS TO CANDIDATES:

1. This paper consists of 5 pages including this cover page. Check carefully to make sure your set is complete.
2. There are **4** questions in this paper. Answer ALL questions.
3. Write your answers in the Answer Booklet.
4. Download the files from Brightspace as instructed by the invigilator.
5. Computer laptops/notebooks (with no internet access) are ALLOWED. Language translators are NOT allowed.

There are 4 questions. Answer ALL questions (100 marks).

**Question 1 (25 marks)**

A pointer-based List ADT (Linked list) is given in the folder Q1.

A new element may be inserted into the correct position in the list sorted in ascending order using the sortedInsert() or sortedInsertR() function.

For example, if the linked list has the following values:

0 2 3 4 6 8 9

using the sortedInsert() function to insert 5 would result in the following linked list:

0 2 3 4 5 6 8 9

Duplicate values are allowed in the list.

(a) Implement the **iterative** void List::sortedInsert(Itemtype item) function.

(10 marks)

(b) Implement the **recursive** void List::sortedInsertR(Itemtype item) function.

(10 marks)

(c) State the time complexity of the function in 1(a) and 1(b). Justify your answers.

(5 marks)

**Question 2 (25 marks)**

A simple console application to simulate a short messaging system (SMS) is to be implemented using the Stack ADT given in folder Q2.

Some sample outputs of the application are shown below.

Main Menu

|  |
| --- |
| Main Menu  -------------------  [1] Compose  [2] Undo  [3] Clear  [0] Exit  -------------------  Enter your option:\_ |

Option 1 : Compose

|  |
| --- |
| Option 1 : Compose  Hello how are yoiu |

Note : text underlined are entered by user.

Option 2 : Undo

|  |
| --- |
| Option 2 : Undo  Hello how are yoiu  Enter 1 to undo, 0 to quit: 1  Hello how are yoi  Enter 1 to undo, 0 to quit: 1  Hello how are yo  Enter 1 to undo, 0 to quit: 0 |

Note : user can repeatedly undo by choosing option 1 (or 0 to quit).

Option 1 : Compose

|  |
| --- |
| Option 1 : Compose  Hello how are you? |

Note : new text are underlined.

Option 3 : Clear

|  |
| --- |
| Option 3 : Clear  The text is cleared. |

(a) Implement the compose() function that allows the user to enter text and save to the stack. The function prototype is given below.

void compose(Stack& s)

(10 marks)

(b) Implement the undo() function that allows the user to undo changes. The function prototype is given below.

void undo(Stack& s)

(10 marks)

(c) Implement the clear() function that allows the user to clear the text. The function prototype is given below.

void clear(Stack& s)

(5 marks)

**Question 3 (25 marks)**

The customer service centre of an IT company has decided to implement a queue system to process the service requests of its customers. The specification of the queue is given below. A queue ADT is given in folder Q3.

1. Implement a function, void registerCustomer(Queue& serviceQueue, int& queueNumber), to allow the staff to register a customer. The function should prompt the customer to enter his/her name and add a customer to the queue, serviceQueue with the queueNumber, followed by incrementing the queueNumber for next potential customer.

(10 marks)

1. Implement a function, void nextCustomer(Queue& serviceQueue), to allow the staff to get the next customer in the queue to serve. The function should remove the customer object from the queue and display the queue number of that customer.

(5 marks)

1. Implement a function, void displayCount(Queue& serviceQueue),to allow the staff to compute and display the total number of customers left in the queue.

(10 marks)

**Question 4 (25 marks)**

1. Consider the following 2 hash functions:

public int hash1(String str)

{

int hash = 0;

for (int i=0; i< str.length(); i++)

hash = hash \* 31 + int(str.at(i));

return hash;

}

public int hash2(String str)

{

int hash = 0;

int skip = Math.max(1, str.length()/4);

for (int i=0; i< str.length(); i+=skip)

hash = hash \* 31 + int(str.at(i));

return hash;

}

Which hash function is better? Explain why.

(6 marks)

(b) Given the following information for a hash table:

The size of the hash table is 7 and chaining is used to resolve collisions.

The hash function used is hash(k) = k % 7.

(i) Draw the hash table after the following sequence of insertions.

10, 36, 17, 19, 24

(8 marks)

(ii) Consider the table size of 7. Give one good reason for picking 7 and one bad reason for picking 7 for the table size.

(6 marks)

(iii) What is the maximum number of entries that can be placed in the hash table? Justify your answer.

(5 marks)

**– End of Paper –**