Part 1 of 5

15 Minutes
09:15 – 9:30

Instructions

1. Be **honest** while solving the paper. It would be better to fail the paper than to pass it by unfair means.
2. You need to **show complete working** for every question, **marks will not be given** in case working is not shown.
3. Solve each part in its given time. Solution of each part has to be submitted in its **given time**. **Late** solutions **will not be accepted!**
4. Write **part number** and **sheet number** on top of each sheet.
5. Take pictures of the solution with camera **flash on**. Blur or unclear solutions will not be accepted.
6. Submit the solution through **WhatsApp** only.
7. Do **not create PDF** files for the solution.
8. Do **not delete** any message, even if you post something irrelevant.
9. You should have a working **mic** and **camera**.
10. Share your live location on WhatsApp (Attach->Location->**Share Live Location**).
11. You will **loose marks** if you do not follow any of the instructions.

Q1. {1.5+1.5+1.5+3.5=8}

a) Write down all the values given in front of your name.

b) What is the largest value in the numbers given in front of your name?

c) What is the smallest value in the numbers given in front of your name?

d) Write all the instructions (one time) on the first sheet.

Reg	Name	Values
01072011002	Ayesha Batool	13,**4**,14,20,8,5,**26**,6
01072011005	Samra Batool	3,5,**22**,6,13,19,**2**,10
01072011008	Laraib Fatima	4,28,**30**,7,8,6,**3**,27
01072011020	Farheen Fatima	20,11,**8**,**30**,18,23,28,22
01072011022	M. Haider Shahab	4,**29**,13,24,23,12,**2**,8

Part 2 of 5

35 Minutes
09:30 – 10:05

Important!

- **Show all the working for all the parts of all the questions, otherwise marks will be not given.**
- **No code is required for any question in this part.**
- **Do not run the given code on a computer.**

Q2. Consider an array based stack ADT implementation and answer the following: {1.5+1.5+6=9}

a) How would the array (used to store values in the stack) look like after pushing all the values given in Q1 in front of your name, in the given order?

b) What will be the index of the largest value (Q1-b)?

c) How would the array in the stack look like after performing the following operations?

```
pop();
pop();
pop();
int x = top();
push(x*2);
push(x+5);
int y = top();
push(x+y);
```

Q3. Consider a queue ADT implemented using linked structures. The front element of the queue is pointed by a pointer *q_front* and rear element by *q_rear*. Each element has two parts, one for storing value (*data*) and other for linking to the next element in the queue (*link*). {1.5+7.5=9}

a) How would the linked structures in the queue look like after pushing all the values given in front of your name in Q1 in the given order?

b) Let's assume that a pointer *ptr_largest* points to the largest element (Q1-b) in the queue and *ptr_smallest* points to the smallest element (Q1-c) in the queue. What will be the result of the following statements? In case of an error, explain it.

i. `q_front->data < ptr_smallest->link->data`

ii. `q_front->link == ptr_smallest`

iii. `ptr_smallest->link->link == q_rear`

iv. `q_front->link->link->data > ptr_largest->data`

v. `ptr_largest->link->data == q_rear`

Part 3 of 5

35 Minutes
10:05 – 10:40

Important!

- **Show all the working for all the parts of all the questions, otherwise marks will be not given.**
- **No code is required for any question in this part.**
- **Do not run the given code on a computer.**

Q4. For this question, assume that you have implemented an unsorted doubly linked list. The front element of the queue is pointed by a pointer *head* and rear element by *tail*. New values are added to the end of the list. The largest element (Q1-b) is pointed by a pointer *ptr_largest* and smallest (Q1-c) by *ptr_smallest*. Each element of the linked list has three parts: value (*data*), pointer to the next node (*next*), and pointer to the previous node (*prev*). {3+6=9}

a) How would the doubly linked list look like after inserting all the values given in Q1 in the given order?

b) How would the doubly linked list look like after executing the following statements. In case of an error, explain it.

```
ptr_smallest->next->prev = ptr_smallest->prev;
ptr_smallest->prev->next = ptr_smallest->next;
ptr_smallest->next = ptr_largest->next;
ptr_smallest->prev = ptr_largest;
ptr_largest->next->prev = ptr_smallest;
ptr_largest->next = ptr_smallest;
```

Q5. Answer this question based on array based sorted list. {1.5+3+1.5+3=9}

a) How would the array in the array based sorted list look like after inserting the values given in Q1?

b) Suppose you want to find the value 13 in the list using binary search. Which values will be compared with 13 before the find function returns true or false? Use integer division, if required.

c) What will be the output of the following code?

```
reset();
int v;
for (int i=0;i<5;i++) {
    v = get_next();
}
cout<<v<<endl;
```

d) How the array in the list would look like after performing the following operations? Show the array after each operation.

```
insert(12);
update(12,17);
erase(12);
```

Part 4 of 5

30 Minutes
10:40 – 11:10

Important!

- **Show all the working for all the parts of all the questions, otherwise marks will be not given.**
- **Code is only required for Q6(b).**
- **Do not run the given code on a computer.**

Q6. Answer this question based on AVL tree. {6+3+6=15}

a) How would an empty AVL tree look like after inserting the values given in Q1 in the given order (one by one)? Show all the intermediate steps while inserting the values.

b) For the AVL tree drawn in part (a), let's assume that a pointer *root* points to the root node. Write code to insert a new node as a right child of the predecessor node (here predecessor node means the predecessor to the root node). Assume any valid value for the new node. Do not use any loop, only use the root pointer to insert the new node. You may use a temporary pointer for creating the new node.

c) Given the following function `magic`, how would the tree in part (a) look like after executing the statement `magic(root->left)`? Please note that the tree might not remain a valid BST/AVL after executing this statement.

```
template<typename T>
void magic(bnode<T> *&ptr) {
    bnode<T> *temp;
    temp = ptr;
    ptr = new bnode<T>;
    ptr->data = 31;
    ptr->left = NULL;
    ptr->right = temp;
}

magic(root->left); //execute this statement only once.
```

Part 5 of 5

30 Minutes
11:10 – 11:40

Important!

- **Show all the working for all the parts of all the questions, otherwise marks will be not given.**
- **No code is required for any question in this part.**
- **Do not run the given code on a computer.**

Q7. Answer this question based on the max-heap structure. Let's assume that the values for the heap are stored in an array named *data*. {3+7=10}

a) How would an empty heap look like after inserting the values given in Q1 in the given order? Show all the intermediate steps.

b) How would the heap look like after performing the following operations? Show all the intermediate steps.

```
pop();
pop();
pop();
push(31);
push(12);
```

Q8. Answer this question based on a hash table implemented using the chaining method. Assume that the hash table size is 3 (i.e., $m = 3$). In case of a collision, the values (keys) are stored in a linked list. How would the hash table look like after inserting the values given in Q1? {6}