Lab 8 - DSA

IT F19 Morning

Question 1:

15 marks

Construct a Binary Search Tree. Having this Basic Function. The Class structure must be as follow. No other public member is allowed. You are required to submit BST.h with following class

```
template<class T>
public:
class BST {                 // 15- number
	class Node {
	public:
		T data;
		Node* left;
		Node* right;
	};
private:
	Node* root;
public:
	BST();		// 0.5- number
	void insert(T val);		// 1.5- number
	void remove(T val);		// 2.5-	number
	void printInOrder()const;	// 1.5- number
	void printPostOrder()const; // 1.5- number
	void printLevelOrder(int k)const;  // 2- number
	void replace(T old_val, T new_val); // 5- number
	~BST ();			// 0.5- number
};
```

Testing of the above class will be done by following main file, I will replace nothing in this file, Zero Marks in case of error in compilation :

```
#include<iostream>
#include"BST.h"
using namespace std;
int main()
{
	BST<int> a;
	a.insert(1);
	a.insert(2);
	a.insert(3);
	a.insert(33);
	a.insert(24);
	a.insert(14);
	a.insert(10);
	a.insert(27);
	a.printInOrder();
	a.remove(22);
	a.printInOrder();
	a.replace(10, 13);
	a.printInOrder();
	a.printLevelOrder(2);
	return 0;
}
```

Question 2: 15- Marks

You are said to store data of Hospitals in a city. For that purpose you are asked to develop a structure of maxHeap using these structures. You have to submit Hospital.h file only

```
class Hospital{ // 6- marks
	string HospitalName;
	string Address;
	string HospitalID;
public:
	Hospital();		// 0.25- marks
	Hospital(string ,string ,string); // 0.25- marks
	Hospital(string HospitalID); // conversion Constructor // 0.5- marks
	string getHospitalName()const; // 0.25- marks
	string getHospitalId()const; // 0.25- marks
	string getAddress()const; // 0.25- marks
	void setHospitalName(string); // 0.25- marks
	void setHospitalId(string); // 0.25- marks
	void setAddress(string); // 0.25- marks

	// Relational operators with respect to Hospital Id
	bool operator<(Hospital)const; // 0.5- marks
	bool operator<=(Hospital)const; // 0.5- marks
	bool operator>(Hospital)const; // 0.5- marks
	bool operator>=(Hospital)const; // 0.5- marks
	bool operator==(Hospital)const; // 0.5- marks
	bool operator!=(Hospital)const; // 0.5- marks
	// output stream operator
	friend ostream& operator<<(ostream&, const Hospital& ref); // 0.5- marks

};
ostream& operator<<(ostream&, const Hospital& ref);

class MaxHeap {	// 9
public:
	class Node {
	public:
		Hospital data;
		Node* left;
		Node* right;
	};
private:
	Node* root;
public:
	MaxHeap();	//	0.5- marks
	bool Insert(Hospital); // 2- marks
	bool Remove(Hospital); // 2- marks
	bool Replace(Hospital old_Hos, Hospital new_Hos); // 3- marks
	Hospital* getHeapArray()const; // 1-mark
	~MaxHeap();	//	0.5- marks
};
```

Testing of the above class will be done by following main file, I will replace nothing in this file, Zero Marks in case of error in compilation :

```
#include<iostream>
#include"Hostpital.h"
using namespace std;

int main()
{
	MaxHeap a;
	a.Insert(Hospital("Meo", "Lahore", "HL1"));
	a.Insert(Hospital("Jinnah", "Lahore", "HL2"));
	a.Insert(Hospital("Civil", "Lahore", "HL3"));
	a.Insert(Hospital("Cardialogy", "Lahore", "HL4"));
	a.Insert(Hospital("Nishtar", "Multan", "HM1"));
	a.Insert(Hospital("Punjab Medical", "Faisalabad", "HF1"));
	a.Insert(Hospital("Allied", "Faisalabad", "HF2"));

	Hospital* ptr= a.getHeapArray();
	for (int i = 0; i < 7; i++)
	{
		cout << i[ptr]<<endl;
	}
	delete[] ptr;
	a.Remove(Hospital("HF2"));
	a.Replace(Hospital("HL4"), Hospital("THQ", "Burewala", "HB1"));
	ptr = a.getHeapArray();
	for (int i = 0; i < 6; i++)
	{
		cout << i[ptr] << endl;
	}
	delete[] ptr;
	return 0;
}
```

Question 3:

15 marks

Make a graph data structure for courses. Which have following functions

void InsertVertex(Course) // 2-marks

void InsertPair(Course,Course) // 2-marks

bool SearchVertex(Course) // 2-marks

void inorderTreversal // 3-marks

void BSF(Course) // will also print the pre requisite Course of the Passed Course 3-marks

void DSF(Course) // will also print the pre requisite Course of the Passed Course 3-marks

this task is open one you have to implement all these functionalities of graph and work must be class based object oriented otherwise zero mark will be awarded