Name: _____ Seat no. _____ Perm no. : _____

Student to your left: _____ Student to your right _____

Please ask all questions in writing using the scratch sheet.

Part 1: [20 points]

Q1 [5pts]

i. _____Set_____ ii. _____queue_____ iii. _Set, priority_queue_

iv. _Vector, set_ v. _Set,_

Q2 [5pts]
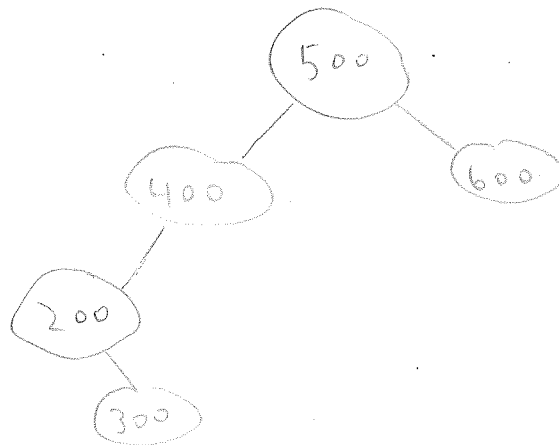
Q3 [10 pts]

i. _____No_____ ii. _____Yes_____ iii. _____Yes_____

iv. _Cannot be determined_ v. _____A_____ vi. _____N/A_____

vii.
A	I	F	D	E	B	C	G	H
0	1	2	3	4	5	6	7	8

viii.
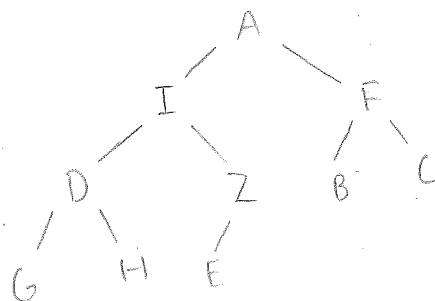
1

Name: _____ Seat no. _____ Perm no. : _____

Student to your left: _____ Student to your right _____

Part 2: [40 points]

Q4 [6pts]

i. Error: the card class can only ^be created using a parameterized constructor

ii. Correct

iii. Correct

iv. Error: the main function cannot directly modify private member variables

v. Error: the member function cannot modify member variables because it is const

vi. Error: s cannot be modified because it is passed in as const

Q5. [4 pts] Implement the overloaded equality operator (==) for the **Card** class as a non member function.

```
bool operator == (const Card& c1, const Card& c2){
    bool sameSuit = c1.getSuit() == c2.getSuit();
    bool sameValue = c1.getValue() == c2.getValue();
    return sameSuit && sameValue;
}
```
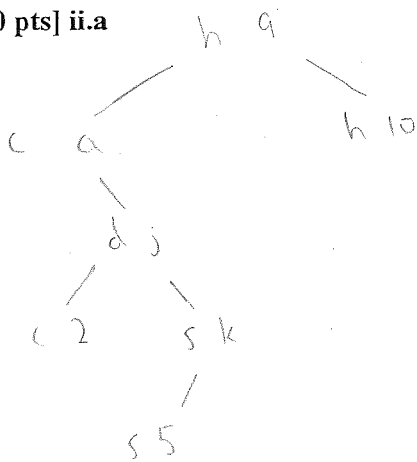
Name: _____ Seat no. _____ Perm no. : _____

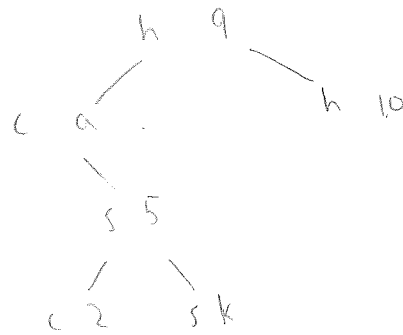Student to your left: _____ Student to your right _____

Q6 i. [10 pts] Implement the overloaded < operator for the **Card** class as a non-member function.

```
bool operator < (const Card& c1, const Card& c2){
    char c1Suit = c1.getSuit(), c2Suit = c2.getSuit();
    int c1Total = 0, c2Total = 0;
    if (c1Suit == 'h')
        c1Total = 10000;
    else   c1Total = (c1Suit - '0') * 30;
    if (c2Suit == 'h')
        c2Total = 10000;
    else   c2Total = (c2Suit - '0') * 30;
    c1Total += c1.getValue();
    c2Total += c2.getValue();
    return c1Total < c2Total;
}
```

Q6[10 pts] ii.a   **ii.b**

Name: _____ Seat no. _____ Perm no. : _____

Student to your left: _____ Student to your right _____

Q6 iii [10 pts] Implement `playgame()`

```
string playgame (vector<Card>& alice, vector<Card>& bob) {
    int ascore = 0, bscore = 0;
    priority_queue <Card, vector<Card>, greater<Card>> pq1;
    priority_queue <Card, vector<Card>, greater<Card>> pq2;
    for (auto item : alice)
        pq1.push (item);
    for (auto item : bob)
        pq2.push (item);

    int i = 0;
    while (!pq1.empty()) {



        if (pq1.top() < pq2.top())
            ascore++;
        else if (p2.top() < pq1.top())
            bscore++;

        i++;

    }




    if (ascore == bscore)
        cout << "Tie" << endl;
    else if (ascore > bscore)
        cout << "Alice" << endl;
    else cout << "Bob" << endl;

}
```

Name: _____ Seat no. _____ Perm no. :_____

Student to your left:_____ Student to your right _____

Part 3: [20 points] Running Time

Q7 [10 pts]

i. $O(1) + (N/2 * \log(N) * O(1)) = N/2 * \log(N) = O(N\log(N))$

ii. $O(1) + (N/2 * N/2 * O(1)) = N/2 * N/2 = O(N^2)$

iii. $O(1) + (\log(N) * O(1)) = O(\log(N))$

iv. $O(1) + (N * \log(N)) + \log(N) = N * \log(N) = O(N\log(N))$

v. $O(1) + (N * \log(N)) + \log(N) = N * \log(N) = O(N\log(N))$

Q8 [5pts]   C, D

Q9 [5pts]

i. ___A___ ii. ___B___ iii. ___B___ iv. ___A___ v. ___C___

Part 4[20 points]:

Q10 [5 pts] int getSum(Node* n) {

```
if (!n)
    return 0;
return (n->data + getSum(n->left) + getSum(n->right));
// the sum of all the nodes in a tree is the value
// of the current node plus the sum of all its nodes
// in the left subtree plus the sum of all its nodes
// in the right subtree
}
```

Name: _____ Seat no. _____ Perm no. : _____

Student to your left:_____ Student to your right _____

Q11 [5 pts] You have an option to implement `linearizeInorder` or `linearizePreorder`

```cpp
void linearizePreorder(Node * r, vector<int>& v){
    if (r){
        v.push_back(r->data);
        linearizePreorder(r->left, v);
        linearizePreorder(r->right, v);
    }
}
```

Name: _____ Seat no. _____ Perm no. :_____

Student to your left:_____ Student to your right _____

```
Q12 [10 pts] bool BST::operator==(const BST& b1, const BST& b2){
        vector <int> w1;
        vector <int> w2;
        linearizeInOrder (b1.root, w1);
        linearizeInOrder (b2.root, w2);
        if (w1.size() != w2.size())
            return false;
        for (int i = 0; i < w1.size(); i++) {
            if (w1[i] != w2[i])
                return false;
        }
        return true;
}
```

Name: _____ Seat no. _____ Perm no. :_____

Student to your left:_____ Student to your right _____

//Use as extra space for any of the questions.