

COSC202 Final 12/10/2025 Instructions

Please answer all questions.

Please write your answers on the test in the spaces provided.

Please put your name and username on all sheets.

I advise that you do you work on scratch paper, and then when you're done, put your answers onto the exam.

Please put your name/username on your scratch paper too, and turn that in.

When you see the classes **Heap**, **BSTree** and **AVLTree**, they have the definitions from class and lab. Specifically:

```
class Heap {
public:
    void    Push(double d);
    double  Pop();
    size_t  Size() const;
    bool    Empty() const;
protected:
    ...
};
```

```
class BSTree {
public:
    BSTree();
    BSTree(const BSTree &t);
    BSTree& operator= (const BSTree &t);
    ~BSTree();

    bool Empty() const;

    bool Insert(const string &key, void *val);
    void *Find(const string &key) const;
    bool Delete(const string &key);

    vector <string> Ordered_Keys() const;
    vector <void *> Ordered_Vals() const;

    int Depth(const string &key) const;
    int Height() const;
protected:
    ...
};
```

```
class AVLTree {
public:
    AVLTree();
    AVLTree(const AVLTree &t);
    AVLTree& operator= (const AVLTree &t);
    ~AVLTree();

    bool Empty() const;

    bool Insert(const string &key, void *val);
    void *Find(const string &key) const;
    bool Delete(const string &key);

    vector <string> Ordered_Keys() const;
    vector <void *> Ordered_Vals() const;

    int Depth(const string &key) const;
    int Height() const;
protected:
    ...
};
```

COSC202 Final Exam - 12/10/2025 - James S. Plank

Question 1 ----- Name/Username: _____

For each procedure, assume that the size of **v1** is **n** elements.
If there is **v2**, then it is **m** elements.
Please enter the best big-O running time of the procedure.
Just write it in an empty space in the box.

The class definitions for Heap, BSTree and AVLTree are on the previous page.

<p>Part A:</p> <pre>void p15f03(AVLTree &v1) { v1.Print(); }</pre>	<p>Part B:</p> <pre>int p43498(const vector <int> &v1) { int k, i, j, rv, d; rv = 0; for (i = 0; i < v1.size(); i++) { for (j = -2; j <= 2; j++) { k = i+j; if (j != 0 && k >= 0 && k < v1.size()) { d = v1[j]-v1[k]; rv += (d*d); } } } return rv; }</pre>
<p>Part C:</p> <pre>void p66a4e(const string &s, void *v, BSTree &v1) { v1.Insert(s, v); }</pre>	<p>Part D:</p> <pre>// In this problem, assume that s is not in the tree v1. void pdd0d9(const string &s, AVLTree &v1) { v1.Delete(s); }</pre>
<p>Part E:</p> <pre>void pcfaee(unordered_set <int> &v1, map <int, int> &v2) { unordered_set <int>::iterator vit; for (vit = v1.begin(); vit != v1.end(); vit++) { v2[*vit]++; } }</pre>	<p>Part F:</p> <pre>void pbe62d(set <int> &v1, unordered_map <int, int> &v2) { set <int>::iterator vit; for (vit = v1.begin(); vit != v1.end(); vit++) { v2[*vit]++; } }</pre>
<p>Part G:</p> <pre>// In this problem, assume that s is in the tree v1. void pf5a0d(const string &s, AVLTree &v1) { v1.Delete(s); }</pre>	<p>Part H:</p> <pre>bool p27964(const string &v1, const string &t) { int i; for (i = 0; i < v1.size(); i++) { if (v1.substr(i)+v1.substr(0,i) == t) return true; } return false; }</pre>
<p>Part I:</p> <pre>vector <double> p3f6ff(Heap &v1) { vector <double> rv; while (!v1.Empty()) rv.push_back(v1.Pop()); return rv; }</pre>	<p>Part J:</p> <pre>void pdfa14(int m, unordered_map <int, int> &v1) { int i; for (i = 0; i < m; i++) v1.insert(make_pair(i, i*i)); }</pre>

Question 2. Name/username: _____

Behold the following program:

<pre>class C09759 { public: int b(); string s; int index; };</pre>	<pre>int C09759::b() { int rv; char c; rv = 0; while (index < s.size()) { c = s[index]; index++; if (c == '(') { rv += (100 * r()); } else if (c == ')') { return rv; } else { rv += (c-'F'); } } return rv; }</pre>	<pre>int main() { C09759 rf; rf.index = 0; cin >> rf.s; cout << rf.b() << endl; return 0; }</pre>
--	--	--

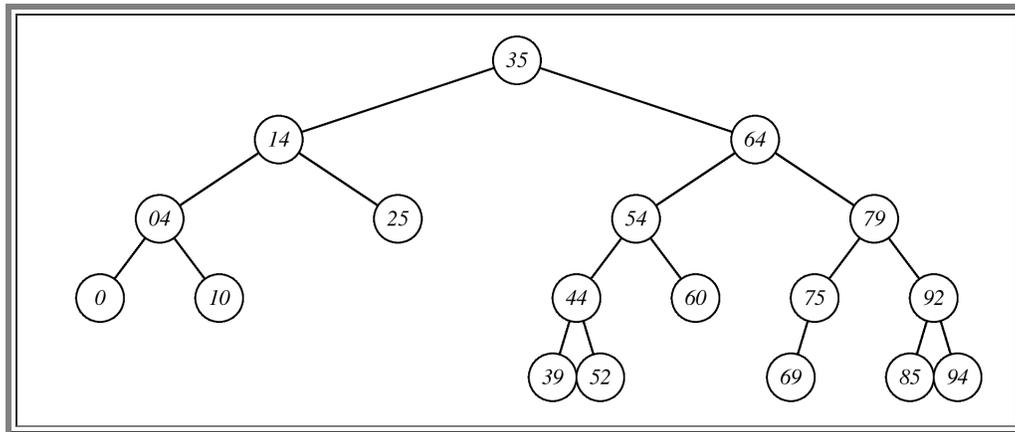
Suppose I compile the program above. For each part below, I give you input to the program, which will be on standard input. Please tell me the output of the program.

Each part is a separate running of the program.

Part	Input	Answer
A	G	
B	(G(HI))	
C	(I)	
D	G(H	
E	((I))	
F	GGG())GGG	
G	(GH)(HG)	

Question 3. Name/username: _____

The following is an AVL tree:



For each part below, I will give you an operation to perform on the tree. Your job is to tell me what rotations will be performed on the tree as a result of the operation. You should either check "No rotations", or fill the "Rotate about ___" boxes with the nodes about which you need to rotate, in the order in which you need to perform the rotations. There will never be more than two rotations necessary.

Answer each part independently, as if the previous part has not been done. In other words, each part works on the same tree, the one pictured above.

5fb4f

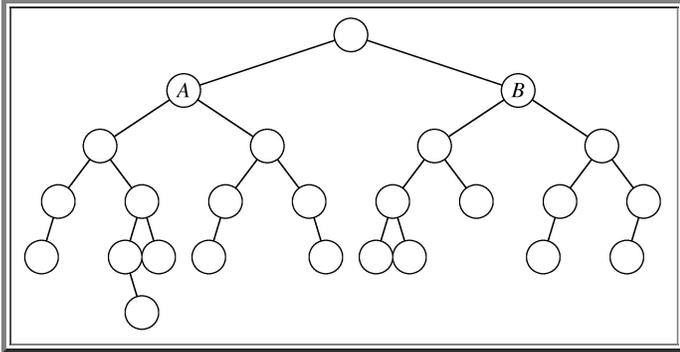
Part	Action	Check this if there are no rotations.	If there is one or more rotations, enter the node about which the first rotation occurs.	If there are two rotations, enter the node about which the second rotation occurs.
A	Insert 13	No rotation __	Rotate about ____	Rotate about ____
B	Insert 48	No rotation __	Rotate about ____	Rotate about ____
C	Delete 25	No rotation __	Rotate about ____	Rotate about ____
D	Delete 60	No rotation __	Rotate about ____	Rotate about ____
E	Delete 79	No rotation __	Rotate about ____	Rotate about ____
F	Insert 19	No rotation __	Rotate about ____	Rotate about ____

Question 4. Name/username: _____

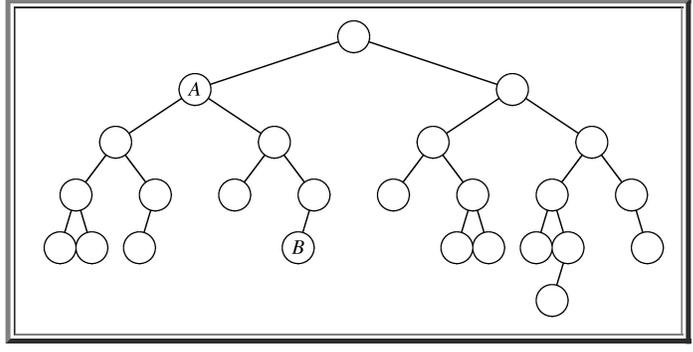
In all of these questions, we are either printing the nodes using a preorder traversal or a postorder traversal. Please circle the correct answer.

5e7fa

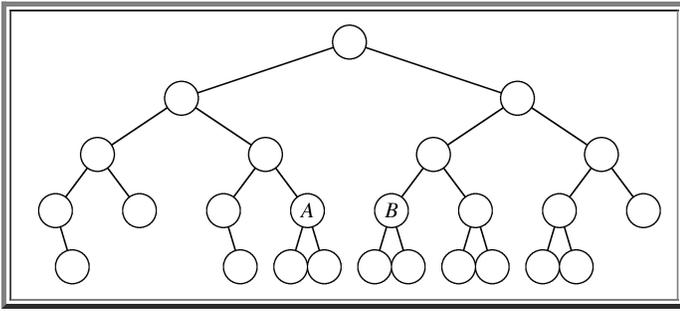
In the following tree:
 Will A be printed before B in a preorder traversal? yes no
 Will A be printed before B in a postorder traversal? yes no



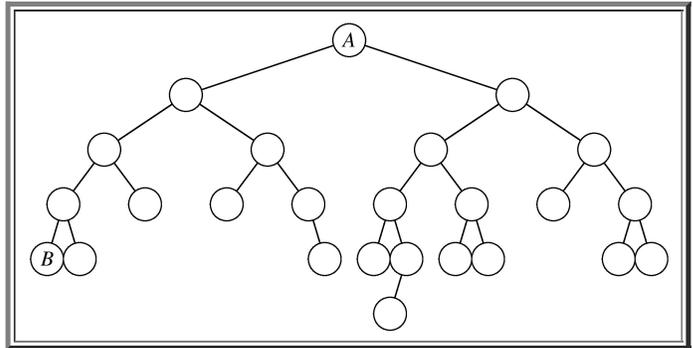
In the following tree:
 Will A be printed before B in a preorder traversal? yes no
 Will A be printed before B in a postorder traversal? yes no



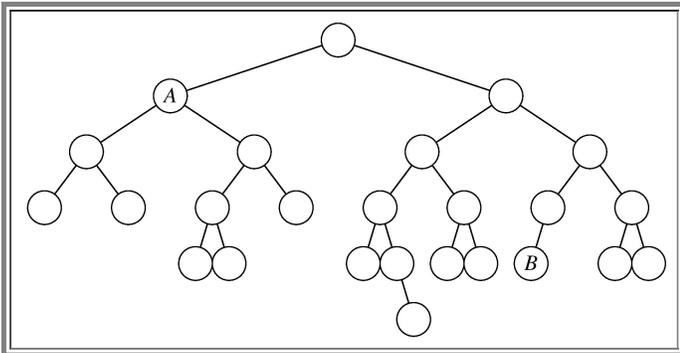
In the following tree:
 Will A be printed before B in a preorder traversal? yes no
 Will A be printed before B in a postorder traversal? yes no



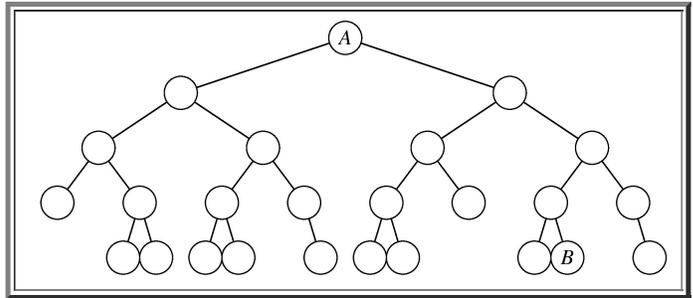
In the following tree:
 Will A be printed before B in a preorder traversal? yes no
 Will A be printed before B in a postorder traversal? yes no



In the following tree:
 Will A be printed before B in a preorder traversal? yes no
 Will A be printed before B in a postorder traversal? yes no



In the following tree:
 Will A be printed before B in a preorder traversal? yes no
 Will A be printed before B in a postorder traversal? yes no



Question 5. Name/username: _____

In all of these questions, when I use the word "value", that means the value stored in the node, and not the node's index. When I use the word "index", that means the index of the vector, and not the value stored in the vector.

ee247

The following is a heap stored as a vector.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	6	9	23	11	38	69	45	30	66	12	67	82	81	94	86	65	97	77	80	90	47	41	99	88

Part A: What is the value of the parent of the node whose value is 65? _____

Part B: What is the value of the left child of the node whose value is 69? _____

Part C: What is the value of the right child of the node whose value is 30? _____

The following is a heap stored as a vector.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	5	9	40	10	22	61	48	44	52	15	28	49	69	99	66	63	72	74	94	68	37	90	95	60	78	81

Suppose that I push the value 34.

Part D: What value will be at index 2? _____

Part E: What value will be at index 6? _____

Part F: What value will be at index 13? _____

Part G: What value will be at index 27? _____

The following is a heap stored as a vector.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	10	6	35	26	7	33	55	42	71	52	9	8	38	44	94	85	62	47	91	87	78	58	83	65	73	61	90	60	95	89

Suppose that I call Pop() on the heap.

Part H: What value will be returned from Pop()? _____

Part I: What value will be at index 0? _____

Part J: What value will be at index 1? _____

Part K: What value will be at index 2? _____

Part L: What will be the index of the node holding the value 89? _____

Question 6. Name/username: _____

Please describe to me how you delete a node in a binary search tree.

Question 7. Name/username: _____

Here is a class definition for a tree node.

```
class Node {  
    public:  
        int value;  
        class Node *left;  
        class Node *right;  
};
```

When we store a tree using this class, we use NULL to represent an empty tree, and we also use NULL when a node does not have a left and/or a right child. Below, write the following procedure, which returns the height of the root node of a tree.

```
int height(Node *t);
```

Do not bother with include files or redefining the class above.

```
int height(Node *t)  
{
```

```
}
```

Question 8. Name/username: _____

Using the same definition of a tree node as in the previous question, please write the following procedure:

```
Node *left_rotate(Node *root);
```

This should modify the tree by rotating about the left child of the root. Return the new root. If you encounter any errors, throw the string "Error".

Note, you should not be calling **new** here to create a new tree -- you are simply modifying the given tree.

As in the previous problem, don't worry about include files or redefining the class.

```
Node *left_rotate(Node *root)
{
```

```
}
```