Karadeniz Technical University
Department of Computer Engineering
Lecturer Ömer ÇAKIR

COM 2005 Data Structures
Resit Exam, 24.01.2019, 13:20
Duration : 61 Minutes

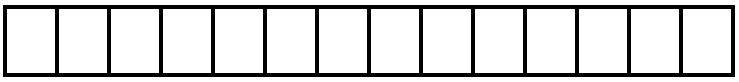
NUMBER : ..... NAME : ..... SIGNATURE :	EXAM GRADE	
	[.....]	.....

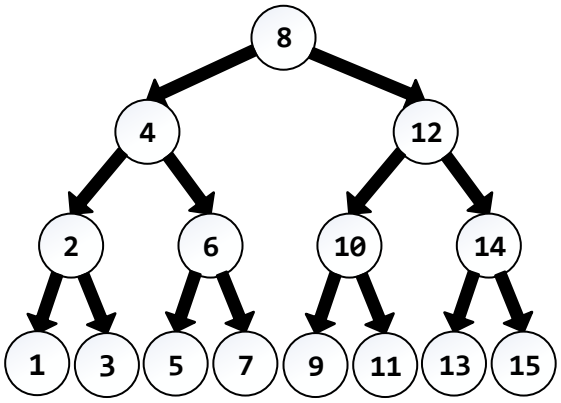
Students have to obey Engineering Faculty Exam Execution Instructions.
Questions are related to 1,4,12 of Program Learning Outcomes

```
void traverse(TreeNode* v)
{
     if (v->left != NULL)
     {
          traverse(v->left);
          cout << v->elem << " ";
     }

     if (v->right != NULL)
     {
          cout << v->elem << " ";
          traverse(v->right);
     }
}
```

1. What is the output of the function **traverse()** that is called with the **root** of the tree below as the argument? **(40P)**





8 7 6 5 4 3 2 1

2. Assume that the numbers above are inserted into a binary tree. Which of the outputs of the `inorder`, `preorder` and `postorder` traversals is different from the other two? **(30P)**

*You'll loose **5P** from wrong answer.*

(A) inorder

(B) preorder

(C) postorder

```
void insertOrdered(string& e, int& i)
{
  DoublyNode* newNode      = new DoublyNode;
  newNode->elem            = e;
  newNode->score           = i;

  DoublyNode* current      = header;

  while (current->next != trailer)
  {
    if (newNode->score >= current->next->score)
      current = current->next;
    else
      break;
  }

  newNode->next     = current->next;
  newNode->prev     = current;
  ..............    = ..............;
  ..............    = ..............;
}
```

3. Complete the function **insertOrdered()** **(30P)**

*You'll loose **5P** from wrong answer.*

(A)
```
newNode->next->prev     = newNode;
current->next->prev     = newNode;
```

(B)
```
newNode->prev->next     = newNode;
current->next           = newNode;
```

(C)
```
current->next->prev     = newNode;
newNode->prev->next     = newNode;
```

(D)
```
current->next           = newNode;
current->next->prev     = newNode;
```

(E)
```
newNode->prev->next     = newNode;
current->next->prev     = newNode;
```