1) A binary search tree is used to locate the no 43
Which of the following prob sequences are possib
and which are not? Explain

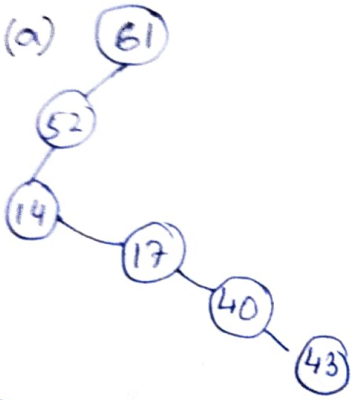
a) 61 52 14 17 40 43

b) 2 3 50 40 60 43

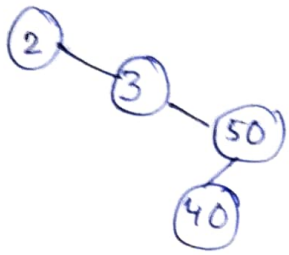
c) 10 65 31 48 37 43

d) 81 61 52 14 41 43

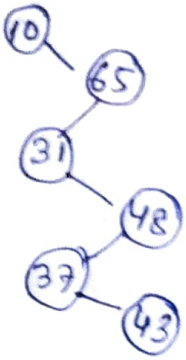
(a)



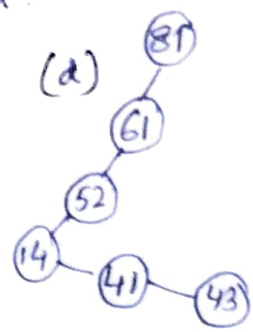
(b)



(c)



(d)



2) A BST contains the value 1, 2, 3, 4, 5, 6, 7, 8. The tre
is traversed in pre-order and the values are
printed out. Which one is a valid output?

a) 5 3 1 2 4 7 8 6

b) 5 3 1 2 6 4 8 7

c) 5 3 2 4 1 6 7 8

d) 5 3 1 2 4 7 6 8

3) Draw the binary tree with node labels a, b, c,
d, e, f, and, g for which the inorder and
post order traversals result in the following
sequences:

INORDER : a f b e d g e

POST ORDER : a f e g e d b

4) Consider the following c function definition:

```
int Trial (int a, int b, int c)
{  if "(a>=b) && (c<b) return b;
   else if (a>=b) return Trail (a, c, b);
}  else return
```

The function Trial :

a) Finds the maximum of a, b, & c

b) Finds the minimum of a, b, & c

c) Finds the middle of a, b, & c

d) None of them.

5) Let LASTPOST, LACTIN, & LAST PRE denote the last vertex visited in a postorder, inorder, & preorder traversal respectively, of a complete binary tree.
Which of the following is always true ?

a) LASTIN = LASTPOST b) LASTIN = LASTPRE

c) LASTPRE = LASTPOST d) None of the above

6) The following postfix expression, containing single digit operands and arithmetic operators + and * ; is evaluated using a stack.
Show the content of the stack, 52 * 34 + 52 * * +

i) After evaluating 52 * 34 +

ii) " " 52 * 34 + 52

iii) At the end of evaluation.

7) An array contains four occurrences of 0, five occurrences of 1, and three " of 2 in any order. The array is to be sorted using swap operations (elements that are swapped need to be adjacent).
What is the minimum no. of swaps needed to sort such an array in the worst case?
What about the total no. of comparisons?

what is the minimum no. of stacks of size n required to implement a queue of size n ?

9) consider the following declaration of a 2D array in C : char a[100][100];

Assuming that the main memory is byte-addressable and the array is stored starting from memory address 0, the address of a[40][50] is :

a) 4005 b) 4050 c) 5040 d) 5400.

10) A single array Arr[1...M] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top-1 and top-2 (top1 < top2) point to the location of the topmost elements in these stacks. If the space is to be used efficiently, the condition for "STACK FULL" is :

a) top-1 + top-2 = M b) top-1 = M/2 or top-2 = M/2

c) top-1 = top2 - 1 d) top1 = M/2 and top-2 = $\frac{M}{2}$ + 1.

11) A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should P point such that both the the operations EnQueue & DeQueue can be performed in constant time ?

12) Post order traversal of a given BST, produces the following sequence : 10 9 23 22 27 25 15 50 95 60 40 29

which one of the following sequences can be result of an in-order traversal of that BST.

a) 9, 10, 15, 22, 23, 25, 27, 29, 40, 50, 60, 95

b) 9, 10, 15, 22, 40, 50, 60, 95, 23, 25, 29, 29

c) 29, 15, 9, 10, 25, 22, 23, 27, 40, 60, 50, 95

13) Consider the following segment of C-code:

```
int j, n;
j = 1;
while (j <= n)
    j = j*2
```

The number of comparisons made in the execution of the loop for any $n > 0$ is -

a) $\lceil \log_2 n \rceil + 1$ b) n c) $\lceil \log_2 n \rceil$ d)

14) Consider the following C program segment.

```
struct CellNode {
    struct CellNode *leftChild;
    int element;
    struct CellNode *rightChild;
};
int DoSomething (struct CellNode *ptr)
{ int value = 0;
  if (Ptr != NULL)
  { if (Ptr->leftchild != NULL)
        value = 1 + DoSomething (ptr->leftchild);
    if (ptr->rightchild != NULL)
        value = max (value, 1 + DoSomething (Ptr->
                                              rightchild));
  }
return (value);
}
```

∴ value returned by the DoSomething when a pointer to the root of a non-empty tree is passed as argument is :-

a) The no of leaf nodes in the tree

b) " " " internal " " "

c) The height of the tree.

15) Consider the datatype struct cellNode as defined in 8.14, together with the following C function

```
int Getvalue (struct cellNode *ptr)
{ int value = 0;
  if (ptr != NULL)
  { if ((ptr->leftchild == NULL) && (ptr->rightchild ==
                                     NULL))
      value = 1;
    else
      value = value + Getvalue(ptr->leftchild)
                    + Getvalue(ptr->rightchild);
  }
  return (value);
}
```

The value returned by the Getvalue when a pointer to the root of a binary tree is passed as argument is :-

a) the no of leaf nodes in the tree

b) " " " nodes " " "

c) the height of the tree.