

1- Given the function f:

```
int f(int n)
{
  int s = 0;
  while(n > 1)
  {
    n = n/2;
    s++;
  }
  return s;
}
```

Which is the complexity of f?

A. $O(n \log n)$ B. $O(n)$ C. $O(\sqrt{n})$ D. $O(\log n)$ E. $O(n^2)$

Ans:

D

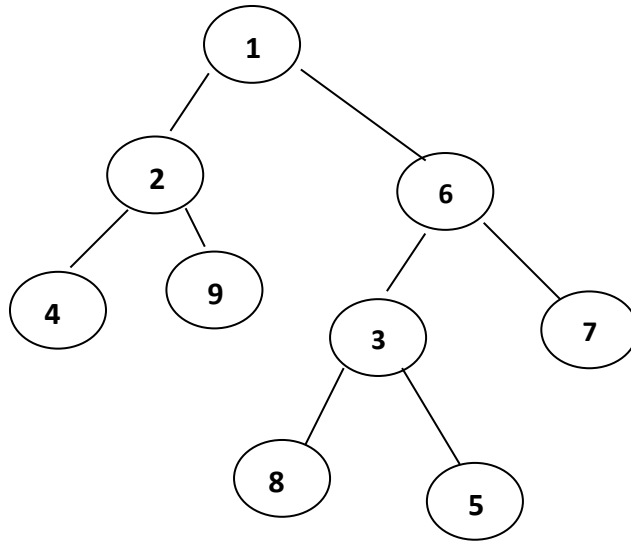
2- Which is the postfix expression of $(12 - a) * (b + 9) / (d * 4)$?

- A. $4 b * d 9 + a 12 - * /$
- B. $/ 12 a - b 9 + d 4 *$
- C. $12 - a * b + 9 / d * 4$
- D. $12 a - b 9 + * d 4 * /$

Ans:

D

3- Consider the binary tree.



What is the pre-order traverse?

- A. 1 2 3 4 5 6 7 8 9
- B. 1 2 4 9 6 3 8 5 7
- C. 4 9 2 8 5 3 7 6 1
- D. 4 2 9 1 8 3 5 6 7
- E. 1 2 6 4 9 3 7 8 5

Ans:

B

4- Consider the linked list 1->2->3->4->5->6. Which is the output of following function assuming that start points to the first node?

```
struct node
{
    int data;
    struct node* next;
};

void fun(struct node* start)
{
    if(start == NULL)
        return;
```

```
printf("%d ", start->data);  
if(start->next != NULL )  
fun(start->next->next);  
printf("%d ", start->data);  
}
```

- A. 1 4 6 6 4 1
- B. 1 3 5 1 3 5
- C. 1 2 3 5
- D. 1 3 5 5 3 1

Ans:

D

5- Insertion of a node, in circular singly linked list, requires modification of?

- A. One pointer
- B. Two pointers
- C. Three pointers
- D. Four pointers

Ans:

B

6- Consider the code

```
int *fun(int *p)  
{  
while(p[2] >= 0) ++p;  
return p;  
}
```

```
void main()  
{
```

```

int *q;
int v[8]={3,2,7,-2,5,6,7,9};
q = fun(v);
printf("%d ", ____Missing_1____);
printf("%d ", ____Missing_2____);
}

```

However, part of the code is missing (indicated by _____). The code is supposed to give the output

7 -2

What can the missing parts be?

- a) Missing_1: *q Missing_2: q[2]
- b) Missing_1: v[4] Missing_2: q[2]
- c) Missing_1: *q Missing_2: q[1]
- d) Missing_1: *(q+1) Missing_2: *(q+2)

Ans:

D

7- What are the time complexities of finding 9th element from beginning and 9th element from end in a singly linked list? Let n be the number of nodes in linked list, and assume that n>9.

- A.) O(n) and O(n) B.) O(1) and O(1) C.) O(n) and O(1) D.) O(1) and O(n)

Ans:

D

8- Consider the linked list 10 12 15 25 30 36 with

```

struct node
{
    int data;

```

```
struct node *next;
};
```

What will be the value of the below expression?

list->next->next->next->data

A.) 12 B.) 15 C.) 25 D.) 30 E.) 36

Ans:

C

9- Consider linked list is used to implement the Stack then which of the following node is considered as Top of the Stack ?

A.) Any Node B.) Last Node C.) First Node D.) Middle Node

Ans:

C

10- When a new element is added in the middle of singly linked list then

- A.) Only elements that appear after the new element need to be moved
- B.) Only elements that appear before the new element need to be moved
- C.) No need to move element
- D.) Only elements that appear after the new element and before need to be moved

Ans:

C

11- What is the output of following function if the start pointing to first node of the linked list:

1->2->3->4->5->6

```
void fun(struct node* start)
```

```
{
```

```

if(start == NULL)
return;
printf("%d ", start->data);
if(start->next != NULL )
fun(start->next->next);
printf("%d ", start->data);
}

```

- A.) 1 4 6 6 4 1 B.) 1 3 5 1 3 5 C.) 1 2 3 5 D.) 1 3 5 5 3 1

Ans:

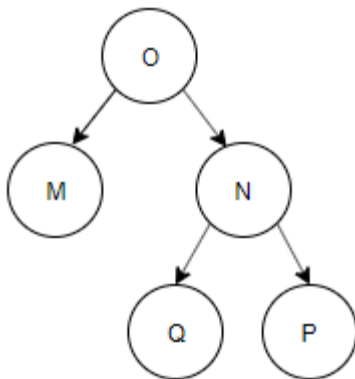
D

12- Which binary tree does yield postorder and inorder traverses as

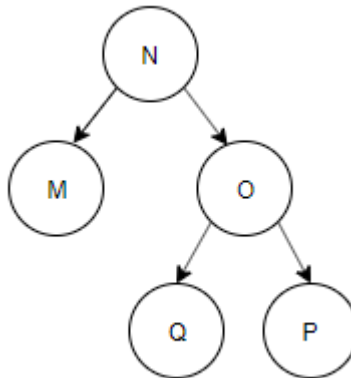
Inorder: N, M, P, O, Q

Postorder: N, P, Q, O, M

A.)

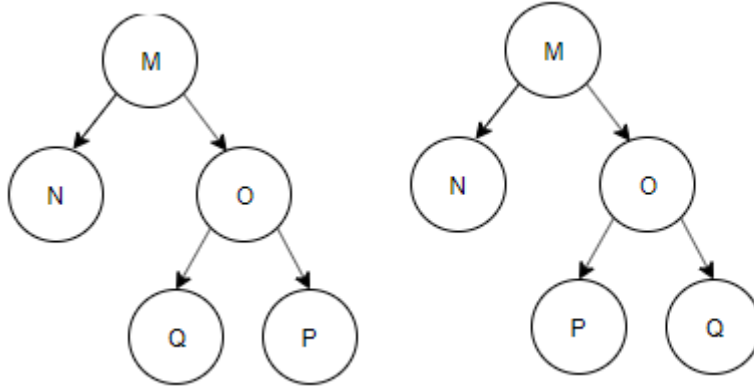


B.)



C.)

D.)



Ans:

D

13- Which of the following properties are obeyed by all three tree traversals?

- a) Left subtrees are visited before right subtrees
- b) Right subtrees are visited before left subtrees
- c) Root node is visited before left subtree
- d) Root node is visited before right subtree

Ans:

A

14- Suppose that T is a binary tree with 14 nodes. What is the minimum possible depth of T?

- a.) 0
- b.) 3
- c.) 4
- d.) 5

Ans:

B

15- Suppose that we constructed a binary search tree for sorting the list of items 14 1 2 5 16 4 in ascending order. Then we remove the root by replacing it with something from the left subtree. What will be the new root?

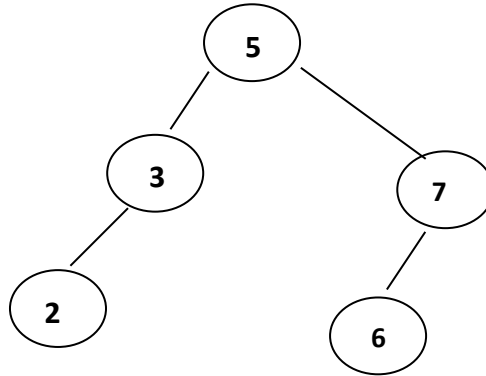
- a.) 1
- b.) 2
- c.) 4
- d.) 5
- e.) 16

Ans:

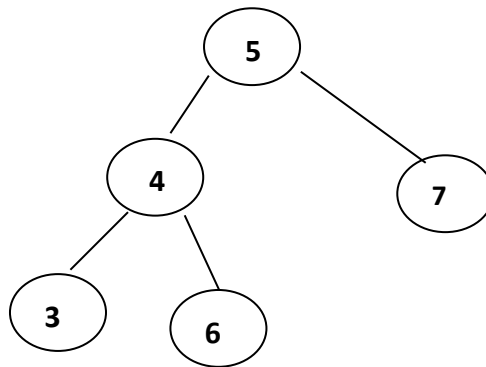
D

16- Which of the following is not a binary search tree?

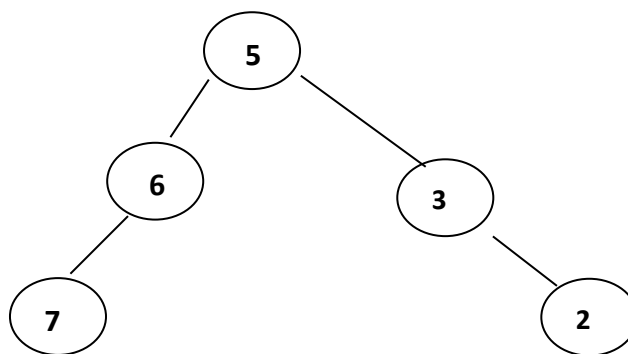
A.)



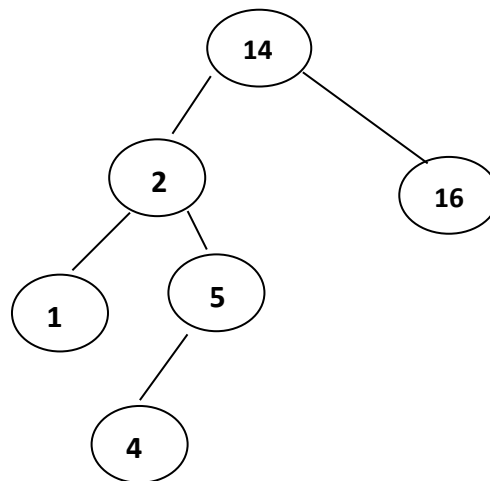
B.)



C.)



D.)



Ans:

B

17- Suppose that we constructed a binary search tree for sorting the list of items 23 11 27 7 25 17 6 14 9 in ascending order. Then we remove the root from the tree. Which of the following (parent, child) pair cannot exist in the tree?

a.) (25,27) b.) (27,11) c.) (11,7) d.) (7,9)

Ans:

B

18- Evaluate the prefix expression: * - + 435 / + 2 4 3

- (a) 8
- (b) 4
- (c) 32
- (d) 16

Ans:

B

19. Which of the following is the prefix notation of the expression $AB+CD-*$?

- (a) $(A+B)*(C-D)$
- (b) $*+AB-CD$
- (c) $+*AB-CD$
- (d) $-CD*+AB$

Ans:

B

20. Choose correct output for the following sequence of stack operations.

```
push (5)
push (8)
pop
push (2)
push (5)
pop
pop
pop
push (1)
pop
```

- A.) 8 5 2 5 1 B.) 8 5 5 2 1 C.) 8 2 5 5 1 D.) 8 1 2 5 5

Ans:

A

21- The post- order traversal of a binary tree is DEBFCA. Find out the pre-order traverse.

- a) ABFCDE b) ADBFEC c) ABDECF d) ABDCEF

Ans:

C

22-The in-order traversal of a binary tree is ABFCD. Find out the pre-order traverse.

- a) ABFCD b) ADBFC c) ABDCF d) None

Ans:

A

23- The best performance occurs for quick sort when the partition splits the array of size n into

- a) $n/2 : (n/2) - 1$
b) $n/2 : n/3$
c) $n/4 : 3n/2$
d) $n/4 : 3n/4$

Ans:

A

24- A machine needs a minimum of 20 sec to sort m elements by Quick sort. The minimum time needed to sort $2m$ elements will be approximately:

- a) $2m+40$ sec
b) $2m$ sec
c) 40 sec
d) $m+20$ sec

Ans:

A

25- Which of the following code segments deletes the element pointed to by q from a doubly linked list? Assume that q does not point to the first or the last element.

- a.) $q \rightarrow \text{left} \rightarrow \text{right} = q \rightarrow \text{right}; q \rightarrow \text{right} \rightarrow \text{left} = q \rightarrow \text{left};$
b.) $q \rightarrow \text{left} \rightarrow \text{right} = q \rightarrow \text{left}; q \rightarrow \text{right} \rightarrow \text{left} = q \rightarrow \text{right};$
c.) $q \rightarrow \text{left} \rightarrow \text{left} = q \rightarrow \text{right}; q \rightarrow \text{right} \rightarrow \text{right} = q \rightarrow \text{left};$
d.) $q \rightarrow \text{left} \rightarrow \text{left} = q \rightarrow \text{left}; q \rightarrow \text{right} \rightarrow \text{right} = q \rightarrow \text{right};$

Ans:

A

26- Which of the following code segments deletes the first element (pointed to by list) from a linear doubly linked list?

- a.) list -> left = list -> right; list -> right = list -> left;
- b.) list = list -> right; list -> left = null;
- c.) list = list -> right; list -> right = null;
- d.) list -> left-> left = list -> left; list -> right-> right = list -> right;

Ans:

B

27- Given the code

```
char (*v)[2], q[4][2]={'C','O','M','P','U','T'}; v=q; v++;  
which of the following is not correct?
```

- a) v[1][1] is 'M'
- b) *v[1] is 'U'
- c) v[-1][0] is 'C'
- d) v[1]-q[1] is 2

Ans:

a

28- Consider the function

```
void fun(char **x) { printf("%s\n", *++x); }
```

which of the following is not correct for the code

```
char *str[3];
```

```
fun(&str[1]);
```

if **XYZ** is printed:

- a) *str[0] may be 'Y'
- b) *str[2] is definitely 'Z'

- c) The string starting at **str[2]** is definitely **"XYZ"**
- d) ***str[1]** may not be **'X'**

Ans:

b

29- Consider the function

```
void fun(char **x) { printf("%s\n", ++*x); }
```

which of the following is not correct for the code

```
char *str[3];
```

```
fun(&str[1]);
```

if **78** is printed:

- a) ***str[2]** may be **'6'**
- b) ***str[1]** is definitely **'7'**
- c) The string starting at **str[1]+1** is definitely **"78"**
- d) **** (str+1)** may be **'6'**

Ans:

d

30- Consider the code

```
char c[] = "COMPUTER";
```

```
struct uuu
```

```
{
```

```
int value;
```

```
char *ptr;
```

```
} q;
```

```
struct uuu *p = &q;
```

```
_____ //line 1
```

_____ //line 2

`printf("%d\n", ++p->value);`

`printf("%c", ++(*(p->ptr)));`

`printf("%c", p->ptr[-1]);`

_____ //line 3

`printf("%c\n", *p->ptr);`

Suppose that the code outputs

8

NOU

What could the code for line1, line2 and line3 be?

- | | | |
|--|---|--|
| a) line 1: <code>p.value = 7;</code> | line 2: <code>p -> ptr = c+2;</code> | line 3: <code>p -> ptr += 2;</code> |
| b) line 1: <code>p -> value = 7;</code> | line 2: <code>p -> ptr = c+2;</code> | line 3: <code>p.ptr += 2;</code> |
| c) line 1: <code>p -> value = 7;</code> | line 2: <code>p -> ptr = c;</code> | line 3: <code>p -> ptr += 2;</code> |
| d) line 1: <code>p -> value = 7;</code> | line 2: <code>p -> ptr = c+2;</code> | line 3: <code>p -> ptr += 2;</code> |

Ans:

d