What will be the space required for this piece of code?

```
int sum (int B[], int n)

{

  int s = 0, j;

  for (j = 0; j < n; j++)

    s = s + B[i];

  return s;

}// sizeof(int) = 2 bytes
```

- 2n + 8
- 2n + 4
- 2n + 2
- 2n

Ans :2n+8

Question 2

Time: 00:00:07

What will be the output of the following pseudo code?

```
For input e = 7 & f = 8.

  work (input e, input f)

  If (e < f)

  return work (f, e)

  elseif (f != 0)

  return (e + work (e, f - 1))

  else

  return 0
```

- 72
- 88
- 56
- 65

Ans 56

work(7,8) will return work(8,7)

work(8,7) will return 8+work(8,6)

work(8,6) will return 8+8+work(8,5)

work(8,5) will return8+8+8+work(8,4)

work(8,4) will return 8+8+8+8+work(8,3)

work(8,3) will return 8+8+8+8+8+work(8,2)

work(8,2) will return 8+8+8+8+8+8+work(8,1)

work(8,1) will return 8+8+8+8+8+8+8+work(8,0)

work(8,0) will return 0

So the result is 8+8+8+8+8+8+8+0=56

Question 3

Time: 00:00:04

What will be the output of the following pseudo code?

```
Input p = 9, w = 6 ,

p = p + 1 ;

w = w - 1 ;

p = p + w

if (p > w)

  print p

else

  print w
```

- 6
- 5
- 10
- 15

Ans 15

After p=p+1, now p=10
after w=w-1, now w=5
after p=p+w, now p=10+5=15
So, at the end
w=5
p=15
As p>w
Hence the output is p or 15

Question 4

Time: 00:00:25

What will be the output of the following pseudo-code?

```
Input t = 6, h = 9 and set x = 0

Integer c

if (h > t)

  for (c = t; c < h; c = c + 1)

    x = x + c

  End for loop

  print x

else

    print error message print x
```

- 21
- 15
- 9
- 6

Ans 21

Prepinsta Explanation User Explanation

In this pseudocode, the for loop operates from c = 6 until c < 9.

In the first iteration we have the value of c = 6, x = 0 + 6 = 6.

In the next iteration we have the value of c = 7, x = 6 + 7 = 13.

In the third iteration we have the value of c = 8, x = 13 + 8 = 21.

The next iteration (c=9) wouldn't be executed since the condition of the loop c < 9 becomes false. And thus the loop terminates.

So the final answer we get is 21.

Question 5

Time: 00:00:08

What will be the output of the following pseudo code?

```
integer i

set i=3

do

print i+3

i=i-1

while( i not equals 0)

end while
```

- 6 5 4
- 6 5 6
- 5 5 5
- 6 6 6

Ans 6 5 4

Skip

PrepInsta Explanation User Explanation

Step 1:
It will print i+3, here i value is 3. So i+3 is 6. On the next line, i will be decremented by 1. Then checking the conditions in do-while() i!=0. Here updated i value is 2 (2!=0),so condition is true. The loop continues.

Step 2:
It will print i+3, here updated i value is 2. So i+3 is 5. On the next line i will be decremented by 1. Then checking the conditions in do-while() i!=0. Here updated i value is 1 (1!=0),so condition gets true. The loop continues

Step 3:
It will print i+3, here updated i value is 1. So i+3 is 4. On the next line i will be decremented by 1. Then checking the condition in do while() i!=0. Here updated i value is 0 (0!=0),so condition gets false. Thus the loop gets terminated!

Question 6

Time: 00:00:04

What will be the output of the following pseudocode for input a = 30, b = 60, C = 90?

```
Integer a, b, c, sum

Read a, b, c

Set sum = a + b + c

if ((sum EQUALS 180) and (a NOT EQUALS 0) and (b NOT EQUALS 0) and (c NOT EQUALS 0))

  Print " Success"

Otherwise

  Print "Fail"

End if
```

- success
- fail
- compilation error
- None of the mentioned

Ans : Success

- a = 30, b = 60, C = 90sum=180((sum EQUALS 180) and (a NOT EQUALS 0) and (b NOT EQUALS 0) and (c NOT EQUALS 0))

  So, (true) and (true) and (true) and (true)

  So, output will be "success"

Question 7

Time: 00:00:03

What will be the output of the following pseudocode for a = 2, b = 6?

```
Integer funn(Integer a, Integer b)

        if(a > 0)

                if(b > 0)

                        return a + b + funn(a + 1, 0) + funn(a + 2, 0) + funn(a + 3, 0)

                End if

        End if

        return a + b

End function funn()
```

- 17
- 21
- 20
- 8

Ans : 20

a=2,b=6

Since both a,b is greater than 0. Hence funn will return 2+6+funn(3,0)+funn(4,0)+funn(5,0)

funn(3,0) will return a+b=3+0

funn(4,0) will return a+b=4+0

funn(5,0) will return a+b=5+0

So, 2+6+3+4+5=20

hence output will be 20

Question 8

Time: 00:00:02

What will be the output of the following pseudocode?

```
Integer count

        for (each count from 0 to 9)

                print "#"

                if (count > 6)

                        CONTINUE

                print count

        End for
```

- #0#1#1#2#3#4#5#
- 0#1#1#2#3#4#5#6##
- #0#1#2#3#4#5#6
- #0#1#1#2#3#4#5#6#7#8#9#10

Ans

- #0#1#2#3#4#5#6

- For loop will iterate from count=0 to count=6Count=0: print "#", print count, so 0 will be printed

  Count=1: print "#", print count, so 1 will be printed

  Count=2: print "#", print count, so 2 will be printed

  Count=3: print "#", print count, so 3 will be printed

  Count=4: print "#", print count, so 4 will be printed

  Count=5: print "#", print count, so 5 will be printed

  Count=6 :print "#", print count, so 6 will be printed

  Count=7: Condition will become become false. Now exit from for loop.

Question 9

Time: 00:00:04

What does the following piece of code do?

```
public void func (Tree root)

{

  func (root.left ());

  func (root.right ());

  System.out.println (root.data ());

}
```

- Preorder traversal
- Inorder traversal
- Postorder traversal
- Level order traversal

- Postorder traversal

In a postorder traversal, first the left child is visited, then the right child and finally the parent.

Learn more about Inorder Postorder preorder here

Inorder Preorder postorder example here

Inorder

- Traverse the left sub-tree, (recursively call ***inorder***(root -> left).
- Visit and print the root node.
- Traverse the right sub-tree, (recursively call ***inorder***(root -> right).

Tips to remember –

- Direction : Clockwise direction
- Rule : LCR i.e. Left ,Center(root), Right

Preorder

- Visit and print the root node.
- Traverse the left sub-tree, (recursively call ***Preorder***(root -> left).
- Traverse the right sub-tree, (recursively call ***Preorder***(root -> right)

Tips to remember –

- Direction : Anti-Clockwise direction
- Rule : CLR i.e. Center(root) ,left, Right

Postorder

- Traverse the left sub-tree, (recursively call ***Postorder***(root -> left).
- Traverse the right sub-tree, (recursively call ***Postorder***(root -> right).
- Visit and print the root node

Tips to remember –

- Direction : Anti-Clockwise direction
- Rule : LRC i.e. Left, Right, Center(root)

Question 10

Tree is a binary search tree. Which of the following code will help us to find the minimum element of Tree?

```
a) public void min (Tree root)

{

  while (root.left () != null)

    {

      root = root.left ();

    }

  System.out.println (root.data ());

}
```

```
b) public void min (Tree root)

{

  while (root != null)

    {

      root = root.left ();

    }

  System.out.println (root.data ());

}
```

```
c) public void min (Tree root)

{

  while (root.right () != null)

    {

      root = root.right ();

    }

  System.out.println (root.data ());

}
```

```
d) public void min (Tree root)

{

  while (root != null)

    {

      root = root.right ();

    }

  System.out.println (root.data ());

}
```

- a
- b
- c
- d

Ans:

- a

Pseudocode practice 1.

(1) Space Required for code

Code:- (c)

```
int sum(int B[], int c){
        int s=0; j;
        for(j=0; j<n; j++)
            s = s + B[j];
        return s;
}
```

Solution:-

(1) Parameters: B[], c
B[] → 2n (space)
c → integer is 2 bytes.

(2) Local variable: s & j → int is 2 bytes
4 bytes total

(3) Return address: Typically 2 bytes.

(4) Total space: 2n (array) + 4 (variables) + 2 (return address) = 2n + 8.

M T W T F S S

COMPASS

Date :

(2) pseudocode output:-

Code:-

```
work (input c, input f);
    If (c<f)
        return work (f, c)
    elseif (f != 0)
        return (c + work (c, f-1))
    else
        return 0
```

Input:- c = 7, f = 8

Solution:- i) work (7, 8) → c<f → work(8,7

ii) work(8, 7) → f != 0 → 8 + work (8, 6)

iii) Continue until f = 0:

- work(8, 6) → 8 + work(8,5
- work(8, 0) → 0

iv) Final Sum: 8 added 7 times

→ 8 * 7 = 56

(3) Pseudocode Output:-

p = 9, w = 6

p = p + 1

w = w - 1

p = p + w

if (p > w) Print p else print w

Solution:-

(1) p = 9+1 = 10

(2) w = 6-1 = 5

(3) p = 10 + 5 = 15

(4) 15 > 5 → Print p = 15

M T W T F S S

COMPASS

Date :

(4) pseudocode output

```
l = 6, h = 9, x = 0
if (h > l)
    for (c = l; c < h; c = c+1)
        x = x + c
    print x
else
    print error
```

Solution:-

(1) loop runs for c = 6, 7, 8:
    x = 0 + 6 + 7 + 8 = 21.

(2) output 21.

(5) pseudocode output

```
i = 3
do
    print i + 3
    i = i - 1
while (i != 0)
```

Solution

1. i = 3 → print 3 + 3 = 6
2. i = 2 → print 2 + 3 = 5
3. i = 1 → print 1 + 3 = 4
4. loop ends (i = 0)
5. output: 6, 5, 4.

(6) pseudocode output

M T W T F S S

COMPASS

Date :

a = 30, b = 60, c = 90

sum = a+b+c

if (sum == 180 and a != 0 && b != 0 && c != 0)

Print "success"

else

Print "Fail"

Solution:-

sum = 180, all input are non-zero

→ "Success"

(7) Pseudocode output:-

```
funn(a,b)
    if(a>0 and b>0)
        return a+b+funn(a+1,0)+
            funn(a+2,0)+funn(a+3,0)
    return a+b
```

Input:- a=2, b=6

Solution:-

funn(2,6) → 2+6+funn(3,0)+
funn(4,0)+funn(5,0)

funn(3,0) → 3+0 = 3

funn(4,0) → 4+0 = 4

funn(5,0) → 5+0 = 5

Total = 2+6+3+4+5 = 20

M T W T F S S

COMPASS

Date :

(8) Pseudocode output:-

```
for (cout = 0 to 9)
    print "#"
    if (cout > 0) continue
    print cout.
```

Solution:-

cout = 0 → print # & 0
cout = 1 → print # (continue skip print cout)
cout = 2 → print # (" " " " ")

Output:- # 0 # # # # # # # (only #0 is clear, option are ambiguous)
closest:- #0#1#2#3#4#5#6

(9) Tree Traversal

Java code

```
void func(Node root){
    func(root.left());
    func(root.right());
    System.out.println(root.data());
```

Analysis:- - Post order Traversal → visit left, then right then root

(10) Finding Minimum in BST

a) Traverse left until Null

M T W T F S S

Date :

b) Traverse left without checking Null

c) Traverse right → Find maximum not minimum.

d) " " " without checking Null