

What will be the output of the following C code?

```
1.
2. #include <stdio.h>
3. int main()
4. {
5.     int x = 2, y = 0, z = 3;
6.     x > y ? printf("%d", z) : return z;
7. }
```

1.

- a. 1
- b. Runtime Error
- c. Compile Time Error
- d. 3

What will be the output of the following C Program?

```
1.
2. #include <stdio.h>
3. int main()
4. {
5.     int a = 0, i = 0, b;
6.     for (i = 0; i < 5; i++)
7.     {
8.         a++;
9.         if (i == 3)
10.            printf("Hello World");
11.         break;
12.     }
13.     printf("%d", a);
14.     return 0;
15. }
```

2.

- a. 1
- b. 2
- c. 4
- d. Hello World

What will be the output of the following C code?

```
1.
2. #include <stdio.h>
3. int main()
4. {
5.     int y = 5;
6.     int z = y +(y = 10);
7.     printf("%d\n", z);
8. }
```

3.

- a. 4
- b. 12
- c. 20
- d. Either 12 or 20

What will be the output of following given statements?

1. Declare the integer variables x, y and z
2. if x > y
3. if z > y
4. Display "One".
5. else
6. if z is equal to x
7. Display "Two".
8. else
9. Display "Three".
10. else Display "Four"

- 4.
- a. It will display two if $x > y > z$
 - b. It will display two if $z \leq y$
 - c. It will display four if $x \leq y$
 - d. It will display four if $x > y < z$

Comment on the output of the following C code.

```

1.
2. #include <stdio.h>
3. void main()
4. {
5.     int k = 4;
6.     int *const p = &k;
7.     int r = 3;
8.     p = &r;
9.     printf("%d", p);
10. }
```

- 5.
- a. It will print address of k + address of r
 - b. It will print address of r
 - c. It will print address of k
 - d. Compile Time Error
6. Which of the following applications may use a stack
- a. A parenthesis balancing program
 - b. Syntax analyzer for a compiler
 - c. Keeping track of local variables at run time

- d. All the above
7. Which of the following is a direct search technique
- Binary Search
 - Tree Search
 - Linear Search
 - Hashing
8. From the following which is NOT true with respect to a STACK
- Stack supports LIFO order of deletion of elements
 - Stack is a dynamic set where elements are removed in the reverse order of their insertions
 - It is not possible to insert or remove elements anywhere else except at the top of the stack
 - All the above are correct
9. From the following where queue implementation is useful
- Load Balancing
 - When data is transferred asynchronously between two processes
 - When a resource is shared among multiple consumers
 - All the above
10. Suppose queue is implemented using a linked list and its front node rear nodes are tracked by two reference variables what reference variables will change during an insertion into a NON EMPTY queue
- Front one
 - Rear One
 - Both
 - None
11. What does the run() do in general
1. `int fun(int x, int y)`
 2. `{`
 3. `if (y == 0) return 0;`
 4. `return (x + fun(x, y-1));`
 5. `}`
 6. `int run(int a, int b)`
 7. `{`
 8. `If (b == 0) return 1;`
 9. `return fun(a, run(a, b-1));`
 10. `}`
- Calculate $x+x*y$
 - Calculate y^x
 - Calculate x^y
 - Calculate $x*y$
12. What will be the output of the following pseudocode

1. Input a = 6, b = 9
2. Integer n
3. Set res = 1
4. if (a > 10 && b < a)
5. Print Error Message
6. else
7. for(n=a; n<b; n=n+1)
8. res = res * n
9. End for loop
10. Print res

Note: '&&' operator returns true if and only if both expressions inside **if()** are true

```

1.
2. if(6>5 && 12<10)
3.     print true
4. else
5.     print false

```

The output will be 'false' as one expression (12<10) in **if()** is not true.

- a. 336
- b. 124
- c. 42
- d. 256

13. What will be the output of the following pseudocode

1. Input n = 1234
2. Integer q, r, and rn
3. set q=n and rn=0
4. while(q > 0)
5. r = q mod 10
6. rn = rn + r³
7. q = q / 10
8. End while loop
9. Print rn

(Note: mod finds the remainder after division of one number by another. For example "5 mod 2" would evaluate to 1 because 5 divided by 2 leaves a quotient of 2 and remainder of 1)

- a. 100
- b. 36
- c. 321

- d. 10
- e. None

What will be the output of the following algorithm for the input x=22 and y=3?

```
1.
2. 1. Start
3. 2. Declare x, y, m, n
4. 3. Set m = x and n = y
5. 4. while m not equal to n then
6. 5.     if m greater than n
7. 6.         m = m - 1
8. 7.     otherwise
9. 8.         n = n - 1
10. 9.     End if
11. 10. End while
12. 11. print n
```

- 14.
- a. 4
 - b. 12
 - c. 3
 - d. None of the above

15. Consider the following pseudocode

- 1. a: = 1
- 2. b := 1;
- 3. while (a <= 500)
- 4. begin
- 5. a: = 2^a;
- 6. b: = b + 1;
- 7. end

What is the value of b at the end of the pseudocode?

- a. 4
- b. 5
- c. 6
- d. 7

16. What will be the output of following code

1. Input m=9, n=6
 2. m=m+1
 3. n=n-1
 4. m=m+n
 5. if(m>n)
 6. print m
 7. else
 8. print n
- a. 5
 - b. 6
 - c. 10
 - d. 15

What will be the output of the following pseudocode?

```
1.
2. 1. Input f = 6, g = 9 and set sum = 0
3. 2. Integer n
4. 3. if ( g > f)
5. 4.     for(n = f ; n < g ; n = n+1)
6. 5.         sum = sum + n
7. 6.     End for loop
8. 7. else
9. 8.     Print Error Message
10. 9. Print sum
```

17.

- a. 9
- b. 21
- c. 6
- d. 15

18. What will be the output of following pseudocode for a given array a[5]=3, 4, 6, 1, 2 and pos is 2?
(Note: n=size of array i.e. 5 and starting array index is 0)

1. Declare i, j, n, pos
 2. Repeat for j=pos to n-1
 3. Set a[j] = a[j+1] [End loop]
 4. n=n-1;
 5. Display the new array
 6. End
- a. 3 2 4 6 1 2
 - b. 3 4 1 2

- c. 3 4 2 1 2
- d. 3 6 1 2