20 MCQ's on Pseudo Codes

1.What is the output of the pseudo code?

```
#include <stdio.h>
int fun(int x, int y, int z)
{
   return x & y ^ x & z;
}
int main()
{
   printf("%d", fun(7, 5, 3));
}
```

OUTPUT : 6

2. What is the output of the pseudo code?

```
#include <stdio.h>
int fun(int a, int b, int c)
{
   return a ^ b ^ (a & c);
}
int main()
{
   printf("%d", fun(5, 6, 1));
}
```

OUTPUT : 2

3. What is the output of the pseudo code?

```
include <stdio.h>
int fun(int x, int y, int z)
{
```

```
    return (x & y & z) ^ (x | y);
}
int main()
{
    printf("%d", fun(7, 3, 1));
}
```

OUTPUT : 6

4. What is the output of the pseudo code?

```
#include <stdio.h>
int fun(int a, int b, int c)
{
    return (a & (b ^ c)) ^ b;
}
int main()
{
    printf("%d", fun(6, 4, 2));
}
```

OUTPUT : 2

5. What is the output of the pseudo code?

```
#include <stdio.h>
int fun(int a, int b, int c)
{
    return a && b || c;
}
int main()
{
    printf("%d", fun(0, 5, 0));
}
```

OUTPUT : 0

6. What is the output of the pseudo code?

```
#include <stdio.h>
int fun(int x, int y, int z)
{
   return x && y || z && x;
}
int main()
{
   printf("%d", fun(0, 5, 1));
}
```

OUTPUT : 0

7. What is the output of the pseudo code?

```
#include <stdio.h>
int fun(int a, int b, int c)
{
   if(a && b)
      return fun(a-1, b-1, c);
   return c;
}
int main()
{
   printf("%d", fun(1, 2, 5));
}
```

OUTPUT : 5

8. What is the output of the pseudo code?

```
#include <stdio.h>
```

```
int fun(int a, int b, int c)
{
   if(a && (b || c))
      return fun(a-1, b-1, c-1);
   return a + b + c;
}
int main()
{
   printf("%d", fun(2, 1, 0));
}
```

OUTPUT : -3

9. What is the output of the pseudo code?

```
#include <stdio.h>
int fun(int a, int b, int c)
{
   if(a && fun(a-1, b, c))
      return 1;
   return 0;
}
int main()
{
   printf("%d", fun(2, 0, 1));
}
```

OUTPUT : 0

10. What is the output of the pseudo code?

```
#include <stdio.h>
int fun(int a, int b, int c)
{
```

```
    return a && fun(a-1, b, c) || c;
}
int main()
{
    printf("%d", fun(1, 0, 0));
}
```

OUTPUT : 0

11. What is the output of the pseudo code?

```
 #include <stdio.h>

int fun(int n) {
          if(n < 2)
                    return 1;
          return fun(n - 1) + fun(n - 2);
}
int main(){
          printf("%d", fun(5));
          return 0;
}
```

OUTPUT : 8

12. What is the output of the pseudo code?

```
#include <stdio.h>
int main()
{
    int p=5, q=8, r=5;
    if ((9 & p) < r || q > r)
        q = 5 & q;
    q = 11 + p;
    if ((q + p) > (p - q))
        r = (p + p) + q;
    p = (p + r) + q;
    printf("%d", p + q + r);
```

```
    return 0;
}
```

OUTPUT : 89

13. What is the output of the pseudo code?

```
#include <stdio.h>
int main()
{
    int a=4, b=6, c=3;
    if ((a + b) > c)
        a = b + c;
    if ((a - c) < b)
        b = a + c;
    c = a + b;
    printf("%d", a + b + c);
    return 0;
}
```

OUTPUT : 30

14. What is the output of the pseudo code?

```
#include <stdio.h>
int main()
{
    int p=6, q=2, r=9;
    if ((p & q) < r)
        p = q + r;
    if ((p | r) > q)
        q = p - r;
     r = p + q;
    printf("%d", p + q + r);
```

```
    return 0;
}
```

OUTPUT : 26

15. What is the output of the pseudo code?

```
#include <stdio.h>
int main()
{
    int x=8, y=4, z=6;
    if ((x + y) > z)
        x = y + z;
    if ((x - z) < y)
        y = x + z;
    z = x + y;
    printf("%d", x + y + z);
    return 0;
}
```

OUTPUT : 28

16. What is the output of the pseudo code?

```
#include <stdio.h>
int main()
{
    int p=5, q=3, r=4;
   if (p>q && q<r)
        p = q + r;
   if (p>r || q>p)
        r = p + q;
printf("%d", p+q+r);
    return 0;
```

```
}
```

OUTPUT : 20

17. What is the output of the pseudo code?

```
#include <stdio.h>
int main()
{
   int p=12, q=6, r=4;
   if (p>q && q>r)
      p = q + r;
   if (p>r || q>p)
      r = p + q;
   printf("%d", p+q+r);
   return 0;
}
```

OUTPUT : 32

18. What is the output of the pseudo code?

```
#include <stdio.h>
void check()
{
   int a = 3, b = 8;
   if (a < b && b > 5)
      a = a + b;
   if (a < b || b < a)
      b = a + b;
printf("%d", a+b);
}
int main()
{
```

```
    int a = 90;
    check();
     return 0;
}
```

OUTPUT : 30

19. What is the output of the pseudo code?

```
#include <stdio.h>
int main()
{
    int p = 8, q = -22, r = 14;
    if ((q + r) < (q - r))
        p = (p + p) + q;
    r = (5 ^ 1) + q;
    if ((q + r) < (p - q) || r > q)
        q = (5 ^ q);
    else
        q = (q + 9) + p;
    printf("%d", p + q + r);
    return 0;
}
```

OUTPUT : -27

20. What is the output of the pseudo code?

```
#include <stdio.h>
int main()
{
    int p = 5, q = 5, r = 10;
    r = (r + q) / r;
   if ((4 + r > p) || (p - q))
```

```
    {
        p = (r + r) + p;
       if ((p + 9 + r) < (6 + p))
        {
            p = 10 + p;
           if ((q + r) < (5 - q))
                p = p + q;
            p = 8 + q;
        }
    }
    r = (r + r) + q;
   q = (p + 12) + r;
   printf("%d", p + q + r);
   return 0;
}
```

OUTPUT : 36

Question: **Difference between break and continue with Example**

Break Statement:

Explanation:
The break statement is used to terminate the loop completely.
When break is executed, control comes out of the loop immediately, and no further iterations are executed.

Example (C language):

```
#include <stdio.h>
int main() {
    for(int i = 1; i <= 5; i++) {
        if(i == 3) {
            break;
        }
        printf("%d ", i);
    }
    return 0;
```

```
}
```
Output:
1 2

Explanation of Example:
The loop starts from 1
When i becomes 3, break executes
The loop stops completely
Numbers after 3 are not printed

Continue Statement:

Explanation:
The continue statement is used to skip the current iteration of the loop.
When continue is executed, control moves to the next iteration of the loop.

Example (C language):
```
#include <stdio.h>

int main() {
    for(int i = 1; i <= 5; i++) {
        if(i == 3) {
            continue;
        }
        printf("%d ", i);
    }
    return 0;
}
```

Output:
1 2 4 5

Explanation of Example:

The loop runs from 1 to 5
When i becomes 3, continue executes
Printing of 3 is skipped
The loop continues with the next iteration