1. **What will be the output of the following pseudocode?**

```
1. Integer a, b, c, d
2. Set a = 10, b = 20, c = 30, d = 40
3. a = b * a
4. b = d - c
5. c = b * 2
6. a = a ^ c
7. b = b - 2
8. b = b << 1
9. c = (c & a) + (a << 1)
10. if(c > 5 || b < 10)
11.         d = a + b + c - 5
12. end if
13. d = d + a
14. Print d
```

[Note- &: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
<< is left shift operator, it takes two numbers, left shifts the bits of the first operand, the second operand decides the number of places to shift.
||: Logical OR - The logical OR operator (||) returns the Boolean value TRUE (or 1) if either or both operands are true and return FALSE (or 0) otherwise.
^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bits of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.]

- ○ 927
- ○ 665
- ◉ 911
- ○ 129

2. **What will be the output of the following pseudocode?**

```
1. Integer a, b, c
2. Set a = 2, b = 40, c = 0
3. b = c + 2
4. if(a)
5.         c = 1
6. End if
7. Print a - b + c
```

[Note: If(x) gets executed if the value inside if(), i.e., x is not zero.]

- ○ 4
- ◉ 1
- ○ -2
- ○ 11

3. What will be the output of the following pseudocode?

```
1. Integer a, b, c
2. Set a = 4, b = 1, c = 2
3. if(b ^ (c & a) && a ^ (c & b))
4.         c = a + a
5.         a = c + c
6. Else
7.         c = b + b
8.         b = c + c
9. End if
10.  Print a + b + c
```

[Note-&&: Logical AND - The logical AND operator (&&) returns the Boolean value true (or 1) if both operands are true and return false (or 0) otherwise.
&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bits of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
If(x) gets executed if the value inside if(), i.e., x is not zero.]

- ○ 22
- ○ 34
- ○ 31
- ◉ 25

4. What will be the output of the following pseudocode?

```
1. Integer a, b, c
2. Set a = 10, b = 1, c = 2
3. if(b&c && a&b && a<<1)
4.         c = c ^ a
5.         a = 0
6. Else
7.         c = 0
8.         a = 2
9. End if
10.  Print a + b + c
```

[Note-&&: Logical AND - The logical AND operator (&&) returns the Boolean value true (or 1) if both operands are true and return false (or 0) otherwise.
<< is left shift operator, it takes two numbers, left shifts the bits of the first operand, the second operand decides the number of places to shift.
&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bits of its second operand. If one bit is 0 and the other bit is 1, the corresponding

result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
If(x) gets executed if the value inside if(), i.e., x is not zero.]

- (●) 3
- ( ) 21
- ( ) 1
- ( ) 11

5. What will be the output of the following pseudocode?

```
1. Integer a, b, c
2. Set a = 1, b = 4, c = 2
3. if(1 && 1)
4.         c = (a & b) + (a ^ b)
5.         if(c)
6.                 c = a
7.         End if
8. End if
9. Print c + a + b
```

[Note-&&: Logical AND - The logical AND operator (&&) returns the Boolean value true (or 1) if both operands are true and return false (or 0) otherwise.
&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bits of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
If(x) gets executed if the value inside if(), i.e., x is not zero.]

- (●) 6
- ( ) 7
- ( ) 8
- ( ) 5

6. What will be the output of the following pseudocode?

```
1. Integer a, b
2. Set a = 3, b = 3
3. a = b
4. b = a
5. if(2 ^ 1 ^ 3)
6.         a = a +1
7. Else
8.         b = b - 1
9. End if
10.  Print a + b
```

[Note- ^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bits of its second operand. If one bit is 0 and the other bit is 1, the corresponding

result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
If(x) gets executed if the value inside if(), i.e., x is not zero.]

- ○ 7
- ◉ 5
- ○ 6
- ○ 4

7. What will be the output of the following pseudocode?

```
1. Integer a, b , c
2. Set a = 1, b = 2, c = 5
3. if(a mod 1 && a^1)
4.         b = b - c
5. End if
6. if(a mod 1 || 1&a )
7.         c = c + a
8. End if
9. Print a + b + c
```

[Note-&&: Logical AND - The logical AND operator (&&) returns the Boolean value true (or 1) if both operands are true and return false (or 0) otherwise.
||: Logical OR - The logical OR operator (||) returns the Boolean value TRUE (or 1) if either or both operands are true and return FALSE (or 0) otherwise.
mod finds the remainder after the division of one number by another. for example, the "5 mod 2" would evaluate to 1 because 5 divided by 2 leaves a quotient of 2 and a remainder of 1.
&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bits of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.
If(x) gets executed if the value inside if(), i.e., x is not zero.]

- ○ 19
- ◉ 9
- ○ 13
- ○ 8

8. What will be the output of the following pseudocode?

```
1. Integer a, b, c
2. Set a = 2, b = 4, c = 2
3. b = a + 1
4. a = c + 1
5. c = b + 1
6. if(a + 2)
7.         if(b + 2)
8.                 a = b + 2
9.         End if
```

```
10.         b = c + 2
11.         if(c + 5)
12.                 a = b + 2
13.         End if
14.  End if
15.  Print a + b + c
```

[Note: If(x) gets executed if the value inside if(), i.e., x is not zero.]

- (●) 18
- ( ) 13
- ( ) 22
- ( ) 26

9. What will be the output of the following pseudocode?

```
Integer pp, qq, rr
Set pp = 3, qq = 10, rr = 13
for(each rr from 5 to 6)
   pp = (9 + 11) + rr
End for
for(each rr from 2 to 3)
   pp = (pp & qq) + pp
End for
Print pp + qq
```

[Note- &: bitwise AND-The bitwise AND operator (&) compares each bit of the first opera corresponding bit of the second operand.
If both bits are 1, the corresponding result bit i Otherwise, the corresponding result bit is set to 0.]

- ( ) 041
- (●) 46
- ( ) 054
- ( ) 52

10. What will be the output of the following pseudocode?

```
Integer pp, qq, rr
Set pp=3, qq=11, rr=15
for(each rr from 4 to 8)
   qq = rr ^ qq
End for
for(each rr from 2 to 4)
   qq = rr + qq
End for
Print pp + qq
```

[Note- ^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bits of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.]

- ( ) 21

- ○ 34
- ○ 10
- ◉ 15

11. What will be the out of the following pseudocode?

```
Integer p, q, r
Set p = 3, q = 8, r = 4
for(each r from 5 to 9)
   q = (q + q) ^  r
   if(q<p OR (q+r) < (p-q))
      q=(q + q) + q
   End if
End for
Print p + q
```

[Note is the bitwise exclusive OR operator that compares each bit of its first corresponding bit of its second operand.
If one bit is 0 and the other bit is 1, t result bit is set to 1. Otherwise, the corresponding result bit is set to 0]

- ○ 362
- ◉ 360
- ○ 358
- ○ 368

12. What will be the output of the following pseudocode?

```
Integer a, b, c
Set a = 2, b = 7, c = 5
for(each c from 3 to 6)
   b = c + b
   if(b + c – a) < (a + b))
      b = 6 + a
   End if
End for
Print a + b
```

- ○ 31
- ○ 36
- ◉ 25
- ○ 18

13. What will be the output of the following pseudocode?

```
Integer a, b
Set b = 320
for (a = 1 to 3)
   b = b / 10
   a = a + b + 5
```

end for
Print a

- ○ 328
- ○ 13
- ○ 26
- ◉ 39

14. What will be the output of the following pseudocode?

```
1. Integer n, beg, end
2. Set beg = 5, end = 7, sum = 0
3. if(beg > end)
4.         Print sum + 1
5. else
6.         for(n = end; n >= beg; n=n-1)
7.                 sum = sum + n
8.                 n = n - 1
9.         End for loop
10.  Print n
```

- ◉ 3
- ○ 7
- ○ 6
- ○ 9

15. What will be the output of the following pseudocode?

```
1. Integer num, x, y, count
2. Set num = 85, count = 0
3. x = num << 1
4. y = x ^ num
5. y = y + 1
6. while((y / 2) NOT EQUALS 0)
7.         if(y MOD 2 NOT EQUALS 0)
8.                 count = count + 1
9.         else
10.                y = y / 2
11.        end if
12.  end while
13.  if(count)
14.        Print “0”
15.        Print y
16.  Else
17.        Print “1”
18.        Print x
19.  end if
```

- ○ 1 125

- ○ None of the mentioned options
- ○ 0 170
- ◉ 1 170

16. What will be the output of the following pseudocode for n = 5?

```
1. Integer i, j, n
2. Read n
3. for(each i from 1 to n)
4.         for(each j from 1 to i)
5.                 Print i
6.         End for
7.         Go to New line
8. End for
```

- ◉ 122333444455555
- ○ 123456789101112131415
- ○ 112123123412345
- ○ None of the mentioned options

17. What would be the output of the following pseudocode for a=2, b=3?

```
doSomthing (integer a, integer b)
If (b EQUALS 1)
   return 0
else
   return a + doSomething(a, b-1)
End function doSomething()
```

- ○ 4
- ◉ 2
- ○ 3
- ○ 1

18. What will be the output of the following pseudocode for input =5?

```
Integer fun(integer n)
   If (n IS EQUAL TO 0)
      return 0
   otherwise if (n is equal to 1)
      return 1
   otherwise
      return (n * n + fun(n-2))
End function fun()
```

- ◉ 35
- ○ 40
- ○ 45
- ○ 25