1. **What will be the output when you execute the following C code?**

```
#include
int main(void)
{
   float f=0.7;
   if(f)
      printf("i am less");
   else
      printf("i am equal");
}
```

- (●) i am less
- ( ) i am less i am equal
- ( ) i am equal
- ( ) None of the above

2. What would the following program produce when executed?

```
#include <stdio.h>
int main()
{
    int ch=2;
    switch (ch)
    {
        case 1:
            printf("1\n");
            break;
        case 2:
                printf("2\n");

        default:
            printf("default\n");
    }
}
```

- ( ) default
- ( ) 2
- ( ) 1
- (●) 2

default

3.

Predict the output of the following code snippet.

```
#include<stdio.h>
int main()
{
    if(-10)
    {
        printf("Hai");
    }
    else
    {
        printf("Bye");
    }
    return 0;
}
```

- ○ Bye
- ○ Compile time error
- ◉ Hai
- ○ HaiBye

4. **Predict the output of the below C program?**

```
#include <stdio.h>
int main()
{
    int x = 3, y = 5, z = 7;
    if (x > 2)
        if (y > 4)
            printf("A");
        else if (z < 8)
            printf("B");
        else
            printf("C");
    else
        printf("D");
}
```

- ○ B
- ◉ A
- ○ C
- ○ D

5.

What is the output of the following C code?

```
#include <stdio.h>
int main()
{
    if (10 > 2 * 4) {
        if (2 > 7 - 8) {
            if (3 == 6 - 3) {
                if (180 % 171 == 9) {
                    if (20 + 2 == 144 / 5) {
                        printf("A");
                    }
                    else {
                        printf("B");
                    }
                }
                else {
                    printf("C");
                }
            }
            else {
                printf("D");
            }
        }
        else {
            printf("E");
        }
    }
    else {
        printf("F");
    }
}
```

- (●) **B**
- ( ) **A**
- ( ) **C**
- ( ) **E**

6.

What will be the output of the following code?

```
#include <stdio.h>
int main()
{
    int n = 3;
    if (2 > 1)
    {
        n += 10;
    }
    if (1 < 2)
    {
        n -= 7;
    }
    if (4 == 7)
    {
        n *= 3;
    }
    n *= 4;
    if (!(4 == 5))
    {
        n++;
    }
    else {
        n--;
    }
    printf("%d", n);
}
```

- ○ 23
- ○ 73
- ○ 74
- ◉ 25

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

```
Integer a, b
Set a = 10
Set b = a + a
if (b > a && 0)
   b = b - a
   b = b mod a
End if
if (b > a || 0)
   b = b + a
   b = b + a
End if
Print b
```

[Note- 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]

[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]

- ○ 50
- ○ 10
- ◉ 40
- ○ 20

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

```
Set length = 5
Set breadth = 7
Set area = length * breadth
Set perimeter = 2 * (length + breadth)
if area > perimeter then
display "Area is greater than perimeter."
else
display "Area is lesser than perimeter."
end-if
```

- ○ Area is lesser than perimeter
- ◉ Area is greater than perimeter
- ○ Error
- ○ Not answered

9.

```
What will be the output of the following pseudocode?
Integer p,q,r
Set p=2,q=7,r=-1
p=p+q+r-q
q=p+r-q
if(p>q)
        print "good bye”
else
        print “take care”
```

- ○ good bye take care
- ◉ good bye
- ○ take care
- ○ None

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

```
int i = 5, j = 7
if(i + j > 5)
  j = i + 2
  if(j < 5)
    print i
  else
```

```
    print j
else
  print i + 1
```

- ○ 12
- ○ 5
- ◉ 7
- ○ 6

11. What will be the outputs for the given inputs if they were went through the following pseudocode?

```
Integer n1, n2, n3, a
n1 = a MOD 10
n2 = a MOD 2
n3 = a/100
if (n1 + n2 > n3)
   print “inside 1st if”
else if (n1 + n2 + n3 > n3 + 3)
   print “inside 2nd if”
else if ((n1 + n2)/n3 EQUALS 0)
   print “inside 3rd if”
else
   print “Last if”
```

Inputs

1. a = 987
2. a = 341
3. a = 247

- ○ 1 – inside 2nd if
  2 – inside 3rd if
  3 – last if
- ○ 1 – last if
  2 – inside 3rd if
  3 – inside 2nd if
- ◉ 1 – inside 2nd if
  2 – inside 3rd if
  3 – inside 1st if
- ○ 1 – inside 1st if
  2 – inside 2nd if
  3 – inside 3rd if

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

```
Integer p, q, r
Set p = 1, q = 5, r = 9
q = q + p
if((q + p) > (r - q))
   p = 5 + q
```

End if
Print p + q + r

- ○ 31
- ◉ 26
- ○ 24
- ○ 37

13. What will be the output of the following pseudocode?
Integer pp, qq, rr
Set pp = 6, qq = 4, rr = 4
if((5 - pp + qq) > (qq - rr))
  rr = (rr + qq) + pp
End if
pp = (7 + 2) + pp
Print pp + qq + rr

- ○ 36
- ○ 33
- ○ 43
- ◉ 23

14. What will be the output of the following pseudocode?
Integer p, q, r
Set p = 0, q = 6, r = 5
if((1 + r + p) > (p + q))
  p = 5 + r
  p = (9 + 12) + q
End if
print p + q + r

- ◉ 11
- ○ 29
- ○ 12
- ○ 5

15. What will be the output of the following pseudocode?
Integer p, q, r
Set p = 3, q = 4, r = 6
p = 3 + r
p = p + q
if((r – q + p) < (p + r))
  q = 2 & r
  q = q + p
End if
Print p + q + r
[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.]

- (●) **34**
- ( ) **18**
- ( ) **29**
- ( ) **37**