GATE

Computer Science & Information Technology

C - Programming

DPP: 2

Data Types & Operators

Q1 What will be the output of the following code?

```
#include <stdio.h>
int main() {
  int x = 10, y=10, z=20;
   y = x++ + ++y + --z;
   printf("%d\n", y);
   return 0;
}
```

(A) 40 (B) 31
(C) 32 (D) 33

Q2

```
#include <stdio.h>
int main()
{
   int x =12, y, b=24;
    y  = x++;
   y = ++x;
   y = x++*b;
   y  = x--;
   y = --x;
   y = x--*b;
   y = --x*b;
   printf("%d",x+y);
   return 0;
}
```

The output of the program is _____

Q3 What will the following code print?

```
#include<stdio.h>
int main () {
   int m=90, k=30;
   int n, n1;
   n=++m + ++k;
   n1=m-- + --k ;
   n--;
   --n1;
   n-=n1;
   printf("%d", n+k);
   return 0;
}
```

(A) 90 (B) 30
(C) 91 (D) 31

Q4 What will be the output of the following code?

```
#include <stdio.h>
int main()
{
  unsigned int a = 45, b = 35;
  int result = (a & b) | (a ^ b);
  printf("%d\n", result);
  return 0;
}
```

(A) 43 (B) 45
(C) 47 (D) 49

Q5 Consider the following code snippet:

```
#include <stdio.h>
int main()
{
      unsigned int x = 7, y = 25;
  int result = ~(x << 2) & (y >> 2);
  printf("%d\n", result);
  return 0;
}
```

What is the output?

(A) 2 (B) 6
(C) 7 (D) 8

Q6 What will be the result of the following code?

```
#include <stdio.h>
int main()
{
  int a = 100, b = 45;
int result = (a ^ b) << 3;
printf("%d\n", result);
return 0;
```

```
}
```

(A) 440 (B) 360

(C) 584 (D) 784

Q7 Consider the following code:

```
#include <stdio.h>
int main()
{
  unsigned int x = 34, y = 15;
  int result = (x & y) ^ (x | y);
  printf("%d\n", result);
  return 0;
}
```

What will be the output of the program?

(A) 15 (B) 30

(C) 45 (D) 60

Q8 What will be the output of the following code ?

```
#include <stdio.h>
int main()
{
  int a = 10, b = 0, c = – 5;
  int result = a && b || c;
  printf("%d\n", result);
  return 0;
}
```

```
}
```

Q9 Consider the following code snippet:

```
#include <stdio.h>
int main()
{
  int x = 5, y = 10;
  int result = (x > 3 && y < 20) || (x++ > 5);
  printf("%d %d\n", result, x);
  return 0;
}
```

What is the output of the program?

(A) 15 (B) 05

(C) 16 (D) 06

Q10 What will the following code output?

```
#include <stdio.h>
int main()
{
  int a = 4, b = -3, c = 0;
  int result = a || b && c;
  printf("%d\n", result);
  return 0;
}
```

GATE

Answer Key

Q1	(A)	Q6	(C)
Q2	275	Q7	(C)
Q3	(D)	Q8	1
Q4	(C)	Q9	(A)
Q5	(A)	Q10	1

Hints & Solutions

Q1 Text Solution:
The initial values are x = 10, y = 10, and z = 20. In the expression y = x++ + ++y + --z, x++ evaluates to 10 (then x becomes 11), ++y becomes 11, and --z becomes 19. Substituting these values, the expression becomes y = 10 + 11 + 19, resulting in y = 40. The program prints 40, so the correct option is A. 40.

Q2 Text Solution:
Initially the program stores the values x = 12 and b = 24. Through various operations, x is incremented and decremented, while y is calculated as follows: after y = --x * b, the final value of y becomes 264, and x is 11. Substituting these values into printf("%d", x + y), we get x + y = 11 + 264 = 275. Hence, the program outputs 275.

Q3 Text Solution:
Iniatally, m = 90 and k = 30. After evaluating n = ++m + ++k, we get n = 122, and after n1 = m-- + --k, n1 = 121. The subsequent decrements make n = 1 and n1 = 120, and n -= n1 results in n = 1. Finally, n + k = 1 + 30 = 31, so the program prints 31.

Q4 Text Solution:
Initially, a = 45 and b = 35, and evaluates the expression result = (a & b) | (a ^ b). The bitwise AND (a & b) results in 33, and XOR (a ^ b) results in 14. The OR operation on these (33 | 14) gives 47. Therefore, the program prints 47.

Q5 Text Solution:
Initially, x = 7 and y = 25, and the expression result = ~(x << 2) & (y >> 2) is evaluated. First, x << 2 results in 28, and its bitwise NOT gives -29. Then, y >> 2 results in 6. The bitwise AND between -29 and 6 gives the final result 2. Therefore, the program prints 2.

Q6 Text Solution:
given, a = 100 and b = 45, then calculates result = (a ^ b) << 3. The bitwise XOR of a and b results in 73. Then, the result (73) is left-shifted by 3 positions, which shifts the binary value of 73 (01001001) to 584 (01001001000). Therefore, the final result of the expression is 584. The program prints 584

Q7 Text Solution:
initially, x = 34 and y = 15, and the expression result = (x & y) ^ (x | y) is evaluated. The bitwise AND of x and y results in 2, while the bitwise OR of x and y gives 47. Next, the XOR operation between 2 and 47 produces 45. Therefore, the final result of the expression is 45. The program prints 45.

Q8 Text Solution:
initially value of a = 10, b = 0, and c = -5, and evaluates the expression result = a && b || c. First, the logical AND (a && b) is evaluated. Since a is non-zero (true) and b is zero (false), the result of a && b is false (0). Next, the logical OR (||) is evaluated with c. As c is -5 (non-zero, true), the result of b || c becomes true (1). Therefore, the final value of result is 1. The program prints 1.

Q9 Text Solution:
in the start , program initializes x = 5 and y = 10, and the expression result = (x > 3 && y < 20) || (x++ > 5) is evaluated. First, the condition (x > 3 && y < 20) is checked. Since x > 3 (true) and y < 20 (true), the entire condition evaluates to true (1). Due to the short-circuiting property of the || (OR) operator, the second condition (x++ > 5) is not evaluated because the first condition is already true. As a result, the value of x remains unchanged at 5. The final value of result is 1. The program prints 1 5.

Q10 Text Solution:
Initially, a = 4, b = -3, and c = 0. The expression result = a || b && c is evaluated based on operator precedence, where logical AND (&&) is evaluated before logical OR (||). First, b && c is evaluated. Since b = -3 (true) and c = 0 (false), the result of b && c is false (0). Next, the logical OR is

evaluated as a || 0. Since a = 4 (true), the final result of the expression is true (1). Therefore, the program prints 1.