GATE

Computer Science & Information Technology

C - Programming

Function & Storage Class

DPP: 2

Q1 The value printed by the program is__?

```
#include <stdio.h>
int fun(){
   int i = 5;
   int a =  i++;
   printf("%d",a);
   return i;
}
int main(){
   fun();
   return 0;
}
```

Q2 The output of the program is ?

```
 #include <stdio.h>
int fun(){
   int i = 5;
   int a =  i++;
   printf("%d",a);
   return i;
}
int main(){

   printf("%d", fun()+fun());
   return 0;
}
```

Q3 what is the output of the given C-Prog?

```
#include <stdio.h>
void fun(){
   static int x=10;
   x+= 20;
   printf("%d ",x);
}
int main(){
   int a = 10, b=20;
   int c;
   fun();
   fun();
   return 0;
}
```

(A) 10, 20 (B) 30, 40
(C) 30, 50 (D) 30, 30

Q4

```
#include <stdio.h>
int fun(int a, int b, int c){
   a&=10;
   b^=20;
   c<<=3;
   return a+b+c;
}
int main(){
     printf("%d",fun(5,23, 4 ));
 }
```

The value printed is___?

(A) 35 (B) 40
(C) 20 (D) 7

Q5 Consider the following Program

```
#include<stdio.h>
    void fun(){
            int i=1, j=1;
            for(; j; printf("%d%d\t",i,j))
                    j=i++ <= 5;
    }
int main(){
    fun();
}
```

what is the output of above program?

(A) 11 22 33 44 5
(B) 21 31 41 51 61
(C) 21 31 41 51 61 70
(D) Infinite loop

Q6 The value printd by the program is ____?

```
#include<stdio.h>
int fun(long int n){
 unsigned long int i, j=0, sum = 0;
  for( i=n; i>1; i=i/2)
          j++;
 for( ; j>1; j=j/2)
          sum++;
 return sum;
}
int main(){
  printf("%d",fun(128));
}
```

Q7 Number of times print statement executed is____?

```
#include<stdio.h>
void fun () {
for (int i = 1; i <= 10; i++) {
   if (i > 5) {
            break;
   }
   printf("%d\n", i);
   }
}
int main(){
   fun();
}
```

(A) 4 (B) 5
(C) 7 (D) 11

Q8 Consider the following program

```
#include<stdio.h>
int fun() {
  int i,j;
  int count=0;
  for(i =1;i<=3;i++){
    for(j=1;j<=20;j++){
      count++;
          if(j==2) break;
   }

  }
   return count ;
}

int main(){
   printf("%d",fun());
}
```

Output of the program is _____?
(A) 4 (B) 5
(C) 6 (D) 7

Q9 Consider the following program

```
#include<stdio.h>
int fun() {
  int i,j;
  int count=0;
  for(i =1;i<=6;i++){
   for(j=1;j<=20;j++){
     count++;
          if(j==2||j>5) break;
   }

  }
   return count ;
}

int main(){
   printf("%d",fun());
}
```

Output of the program is____?
(A) 14 (B) 15
(C) 16 (D) 12

Q10 Consider the following program

```
#include<stdio.h>
int fun() {
  int i,j;
  int count=0;
  for(i =1;i<=3;i++){
    for(j=1;j<=20;j++){
      count++;
          if(i==2) break;
    }
  }
   return count ;
```

```
}
int main(){
    printf("%d",fun());
}
```

Output of the program is___??

(A) 41 (B) 51

(C) 42 (D) 22

Q11 Consider the following program

```
#include<stdio.h>
int fun() {
  int i,j;
  int count=0;
  for(i =1;i<=6;i++){
    for(j=1;j<=20;j++){
      count++;
          if(i==2||i>=5) break;
    }

  }
   return count ;
}int main(){
    printf("%d",fun());
}
```

Output of the program is___?

(A) 41 (B) 63

(C) 42 (D) 22

GATE

Answer Key

Q1	5
Q2	5512
Q3	(C)
Q4	(A)
Q5	(C)
Q6	2
Q7	(B)
Q8	(C)
Q9	(D)
Q10	(A)
Q11	(B)

Hints & Solutions

Q1 Text Solution:
program initializes i to 5, then assigns i++ to a, meaning a gets 5 while i increments to 6. The printf statement prints a, which is 5. The function returns i, but since the return value is not used in main(), it has no effect on the output. The only printed value is from printf, so the output is 5.

Q2 Text Solution:
The function fun() initializes i = 5, assigns i++ to a, prints 5, and returns 6. Since fun() is called twice inside printf, the first call prints 5 and returns 6, and the second call does the same. The returned values 6 + 6 result in 12, which is printed after both 5s. So, the final output is 5512.

Q3 Text Solution:
The function fun() uses a static variable x, which retains its value across function calls. Initially, x is set to 10 but is only assigned once. In the first call, x += 20 makes x = 30, and it prints 30. In the second call, since x retains its value, x += 20 updates it to 50, and it prints 50. Thus, the final output is 30 50. The correct answer is Option C (30, 50).

Q4 Text Solution:
The function fun() performs bitwise operations on the given inputs. First, a &= 10 results in 0 because 5 & 10 in binary gives 00000000. Then, b ^= 20 gives 3, as 23 ^ 20 in binary results in 00000011. Finally, c <<= 3 shifts 4 three positions left, making it 32. Adding all modified values, 0 + 3 + 32 = 35. Hence, the output of the program is 35, and the correct answer is Option A (35).

Q5 Text Solution:
Initialization:
i = 1, j = 1
The for loop condition is j, and since j = 1, it executes.
Loop Execution:
The loop does not have an initialization or condition part, only an update and body.

First iteration:j = i++ <= 5i = 1, so 1 <= 5 is true (1).
i increments to 2, j = 1.
printf("%d%d\t", i, j); prints 21.

Second iteration:j = i++ <= 5 (2 <= 5 → 1, i becomes 3).
printf("%d%d\t", i, j); prints 31.

Third iteration:j = i++ <= 5 (3 <= 5 → 1, i becomes 4).
printf("%d%d\t", i, j); prints 41.

Fourth iteration:j = i++ <= 5 (4 <= 5 → 1, i becomes 5).
printf("%d%d\t", i, j); prints 51.

Fifth iteration:j = i++ <= 5 (5 <= 5 → 1, i becomes 6).
printf("%d%d\t", i, j); prints 61.

Sixth iteration:j = i++ <= 5 (6 <= 5 → 0, i becomes 7).
printf("%d%d\t", i, j); does not execute because j is now 0, exiting the loop.

Final Output:21 31 41 51 61 70
The correct answer is Option C (21 31 41 51 61 70).

Q6 Text Solution:
The function fun(128) starts by initializing j = 0 and sum = 0. The first for loop iterates while i > 1, continuously dividing i by 2 and incrementing j. Starting from i = 128, it goes through 64, 32, 16, 8, 4, 2, 1, making j = 7 after 7 iterations. This loop effectively counts how many times n can be divided by 2 before reaching 1.In the second for loop, j is again divided by 2 in each iteration

while j > 1, and sum is incremented. Initially, j = 7, so j / 2 = 3 and sum = 1. In the next iteration, j / 2 = 1 and sum = 2, which stops the loop since j is now 1. This loop counts how many times j can be halved before reaching 1.After completing both loops, the function returns sum = 2, which is then printed by printf("%d", fun(128));. Hence, the final output of the program is 2.

Q7 Text Solution:

The function fun() runs a for loop where i starts at 1 and increments up to 10. Inside the loop, the if condition checks if i > 5, and if true, it executes break, terminating the loop.The printf statement executes only when i is ≤ 5, printing values 1, 2, 3, 4, 5, after which i = 6, causing the break statement to stop further execution.Since the printf statement executes for i = 1, 2, 3, 4, 5, it runs exactly 5 times. Therefore, the correct answer is Option B (5).

Q8 Text Solution:

The function fun() initializes count = 0 and contains two nested for loops. The outer loop runs for i = 1 to 3, while the inner loop runs for j = 1 to 20. However, the if(j == 2) break; statement forces the inner loop to exit when j = 2.
For i = 1, j = 1 (count = 1), then j = 2 (count = 2), and break occurs.
For i = 2, j = 1 (count = 3), then j = 2 (count = 4), and break occurs.
For i = 3, j = 1 (count = 5), then j = 2 (count = 6), and break occurs.
Since the inner loop only runs for j = 1 and j = 2 in each outer iteration, count increments 2 times per outer iteration. Given 3 outer iterations, count reaches 6.
Thus, the correct answer is Option C (6).

Q9 Text Solution:

The function fun() initializes count = 0 and contains two nested loops. The outer loop runs for i = 1 to 6, while the inner loop runs for j = 1 to 20. However, the condition if(j == 2 || j > 5) break; forces the inner loop to exit when j = 2 or when j > 5.
The inner loop runs from j = 1 onwards.
At j = 1, count++ occurs.
At j = 2, count++ occurs, and then break stops further execution for that iteration of i.
This happens for all 6 iterations of i, meaning each outer loop iteration contributes 2 to count.
Since the outer loop runs 6 times, the total count is 6 × 2 = 12.Thus, the correct answer is Option D (12).

Q10 Text Solution:

The function fun() initializes count = 0 and has two nested loops. The outer loop runs for i = 1 to 3, and the inner loop runs for j = 1 to 20. However, there is a condition if(i == 2) break;, which stops the inner loop immediately when i == 2.For i = 1, the inner loop runs fully from j = 1 to 20, increasing count by 20. When i = 2, the break executes as soon as j = 1, adding only 1 to count, making it 21. For i = 3, the inner loop again runs from j = 1 to 20, adding 20 more to count.Thus, the total count becomes 41. The function returns this value, which is printed in main(). The correct answer is Option A (41).

Q11 Text Solution:

The function initializes count = 0 and contains two nested loops. The outer loop runs for i = 1 to 6, while the inner loop runs for j = 1 to 20. However, there is a condition if(i == 2 || i >= 5) break;, which stops the inner loop execution when i == 2 or i >= 5.For i = 1, the inner loop runs fully for j = 1 to 20, adding 20 to count. When i = 2, the break executes immediately at j = 1, increasing count by 1. For i = 3 and i = 4, the inner loop runs fully again for j = 1 to 20, adding 40 more to count. When i = 5 and i = 6, the break triggers immediately, adding 1 for each iteration.Thus, the total count is 20 + 1 + 20 + 20 + 1 + 1 = 63. The correct answer is Option B (63).