CSE/IT Batch: English

Subject: Programming in C

Topic: Iterative Statements (Loops-2)

DPP-03

[NAT]

1.
```
#include <stdio.h>
int main()
{
    int a=7, b=8;
    while(++b & a-- )
    {
        printf("HI!");
    }
    return 0;
}
```

The number of times the printf() executed is _____.

[MCQ]

2.
```
#include <stdio.h>
int main()
{
    int a=5, b=10;
    while(++b & a-- )
    {
    switch (b)
      {
        case 0: b=b-1;
        break;
        case 1: b=b-2;
        break;
        default: b=b-3;
        break;
      }
    }
    printf("%d\t%d", a, b);
    return 0;
}
```

The output is-

(a) 4 8 (b) 3 9
(c) 3 10 (d) 4 6

[MCQ]

3.
```
int main()
{
    int a=1, b=2;
    do
    {
        while(b++)
        {
            b=b-a;
            a=a+b;
        }
    }
    while(a++<2);
    printf("%d\t%d", a, b);
    return 0;
}
```

The output is-

(a) 3 0 (b) 4 2
(c) 3 2 (d) 4 1

[MCQ]

4.
```
#include <stdio.h>
int main()
{
int a,b;
a=printf("GATE")>printf("Wallah")?printf("2023"):printf("Hi!!");
b=a-1;
while(a>b)
{
    switch(b)
    {
        case 1: b=--a;
        case 2:b=a--;
        default:b=--a;
    }
}
printf("%d%d", a, b);
return 0;
}
```

The output string is-

(a) GATEWallahHi!!33
(b) GATEWallah202303
(c) GATEWallah202330
(d) GATEWallahHi!!00

[NAT]

5.
```
#include <stdio.h>
int main()
{
    int i=16;
    do
    {
        i=i-2;
        printf("%d",i);
        }
    while(i++);
    return 0;
}
```

The sum of all printed values of i is ____?

[MCQ]

6.
```
#include <stdio.h>
int main()
    {
        int a, b;
        while(a!=b)
            {
                a=a/2;
                b=b*2;
                if(b>a) break;
            }
        return 0;
    }
```

If $a = 2^m$ and $b = 2^n$ where m-n is even and positive, the number of times the loop runs is-

(a) $\frac{m-n}{2}$ (b) $\left[\frac{m-n}{2}\right]+1$

(c) $\frac{n-m}{2}$ (d) $\left[\frac{n-m}{2}\right]+1$

[NAT]

7.
```
#include <stdio.h>
int main()
{
    int x=5, y=10;
    if(printf("GATE")-3){
    while(x--) y=y+x;}
    else y=y-x;
```

```
    return 0;
}
```

The value of y at the end of the program is _____.

[NAT]

8.

```
#include <stdio.h>
int main()
{
    int x=5, y=5;
    while(x-=y++<10){
    printf("GATE WALLAH\n");
    }
return 0;
}
```

The number of times “GATE WALLAH” printed is _____.

Answer Key

1. **(7)**
2. **(b)**
3. **(d)**
4. **(a)**
5. **(105)**
6. **(a)**
7. **(20)**
8. **(4)**

Hints and solutions

1. (7)

a = 7 = 0111 a [7 6 5 4 3 2 1 0]

b = 8 = 1000 b [8 9 10 11 12 13 14 15]

while (++b & a – –)

9 & 7 → true

```
1001
0111
0001
```

printf → executed

10 & 6 → True.

```
1010
0110
0010
```

printf → executed ;

similar explanation –

∴ printf( ) is executed –

for a = 7,6, 5, 4, 3, 2, 1 → Total 7 times

2. (b)

a [5 4 3] b [10 11 8 9]

while (++b & a – –)

11 & 5 → true

```
1011
0101
0001
```

switch (11)

default: b = 11 – 3

9 & 4 → false

∴ Final values of a and b are –

[3 9]

3. (d)

a [1 2 4] b [1 2 2 2 0 1]

```
do
 {
    While (b++)
       {
          // values assigned are 2, 2, 0
          b = b – a = 3 – 1 = 2 // 3 – 3 = 0
          a = a + b = 1 + 2 = 3 // 3 + 0 = 3
        }
    }
```

∴

```
while (a ++ < 2);
```

3 < 2 → false

final values are –

a	b
4	1

4. (a)

```
a = print f ("GATE") > printf ("Wallah" )?printf("2023"): printf("Hi!!");
```

4 > 6 → False

a = 4; Output:

b = a-1; // b = 3 GATEWallahHi!!33

```
while (a > b ){
```

4 > 3

3 > 3 → false

```
switch (b){
```

3

```
case 1 : b = -- a;
```

a [4 3]

```
case 2 : b = a --;
```

b [3 3]

```
default : b = -- a;
    }
}
```

5. (105)

```
Do
{
    i = i – 2;
    printf("%d",j);  //14 13 ….. 0
    }
while (i ++);
```

14
13
.
.
.
$0 \rightarrow$ stop.
Value printed–
14 13 12 11 …… 1 0
Sum of the values–

$$\Rightarrow \frac{14 \times (14+1)}{2}$$

$$\Rightarrow 105$$

6. (a)

a = 1024; b = 64

while (a ! = b)

{

1024 ! = 64 $\rightarrow$ True

256 ! = 256 $\rightarrow$ False

a = a/2; //a = 512, 256

b = b *2; //b = 128, 256

if (b > a) break;

}

$\therefore$ The loop runs for 2 times $\left(\because \frac{10-6}{2} = 2\right)$

7. (20)

printf() prints and returns the number of characters it successfully printed. So, the condition becomes (4-3) i.e 1 which is TRUE.

So, the while loop is executed for x values= 5, 4, 3, 2, 1

y=10+4+3+2+1+0 = 20

8. (4)

x-=y++<10 will make the condition true for x = 4, 3, 2, 1 and y=5, 6, 7, 8.

Hence, printf() will be executed 4 times.