Branch : CSE & IT Batch:English

Programming in C
Recursion

DPP-04

[NAT]

1. Consider the following function:

```
int func(int a)
{
   static int b=1;
   b=b+a;
   if((b%a)%2!=0) return a+func(b+a);
   return b-a;
}
```

The value returned by func(5) is _____.

[MCQ]

2. Consider the following function:

```
void func(int n)
{
   if(n>0){
      func(n-1);
      printf("%d\t", n);
   }
   printf("%d\t", n-1);
}
```

The output printed by func(2) is-

(a) –1 1 0 2 1 (b) –1 1 0 2 1 2

(c) –1 1 0 2 -2 1 (d) –1 1 0 2 1

[NAT]

3. Consider the following function:

```
int func(int n)
{
   if(n>0){
      return func(n/2)+func(n/4)+1;
   }
   return n+1;
}
```

The value returned by func(6) is _____

[NAT]

4. Consider the following function:

```
int func(int n)
{
   if(n>0){
      return 3*func(n/4)+1;
   }
   return n;
}
```

The value returned by func(24) is _____.

[NAT]

5. Consider the following function:

```
int func(int n)
{
   static int k=0;
   if(n>0){
      k++;
      return 2*func(n/2)+k;
   }
   return n+k--;
}
```

The value returned by func(8) is _____.

[MCQ]

6. Consider the following function:

```
int func(int n, int i)
{
   if(n==0) return 0;
   else if(n%2){
      return func(n/2, 2*i)+i;
   }else return func(n/2, 2*i)-i;
}
```

The value returned by func(14, 1) is-

(a) 1 (b) 13

(c) 15 (d) 0

[MCQ]

7. Consider the following function:

```
int func(int n)
{
   static int i=0;
   if(n/2){
      i--;
      return func(n/2)+i;
   }else return i;
}
```

The value returned by func(7) is-

(a) –6

(b) –12

(c) –18

(d) –21

[MCQ]

8. Consider the following function:

```
void display()
{
   static int i;
   if(i<=printf("GATE24")){
      i=i+2;
      display();
   }
}
int main()
{
   int i=0;
   for(i=0;i<3;i++)
   display();
   return 0;
}
```

The number of times printf() executed is-

(a) 6 (b) 5

(c) 7 (d) 9

Answer Key

1. **(11)**
2. **(d)**
3. **(9)**
4. **(13)**
5. **(109)**
6. **(b)**
7. **(a)**
8. **(c)**

Hints and Solutions

1. (11)

func(5):

static int b=1; //static b is initialized to 1.

b=b+a; //b=1+5=6

if((b%a)%2!=0) //(6%5)%2!=0 is TRUE

Line 1: return a+func(b+a);// return 5+func(6+5); func(11) is called. Returns 5+6 i.e **11**.

Line 2: return b-a;

func(11):

static int b=1; //static b is initialized to 1.

b=b+a; //b=6+11=17

if((b%a)%2!=0) //(17%11)%2!=0 is FALSE

Line 1: return a+func(b+a);

Line 2: return b-a; // return (17-11) i.e 6 to Line 1 of func(5);

2. (d)

func(2):

2>0 True

func(1) is called.

printf("%d\t", n); // 2 is printed.

printf("%d\t", n-1); //1 is printed.

func(1):

1>0 True

func(0) is called.

printf("%d\t", n); // 1 is printed.

printf("%d\t", n-1); //0 is printed.

func(0):

0>0 is FALSE

printf("%d\t", n-1); //-1 is printed.

Output: -1 1 0 2 1

3. (9)

func(6):

if(n>0){ //6>0 is TRUE

1. return func(n/2)+func(n/4)+1;//return func(3)+func(1)+1; return 5+3+1=9;

}

func(3):

if(n>0){ //3>0 is TRUE

1. return func(n/2)+func(n/4)+1;//return func(1)+func(0)+1; return 3+1+1=5;

}

func(1):

if(n>0){ //3>0 is TRUE

1. return func(n/2)+func(n/4)+1;//return func(0)+func(0)+1;//return 3;

}

func(0):

return 1;

4. (13)

func(24):

if(n>0){ //24>0 is TRUE

return 3*func(n/4)+1; //func(6) is called. Returns 3*4+1; Returns 13.

}

func(6):

if(n>0){ //6>0 is TRUE

return 3*func(n/4)+1; //func(1) is called. Returns 3*1+1; returns 4;

}

func(1):

if(n>0){ //1>0 is TRUE

return 3*func(n/4)+1; //func(0) is called. Returns 1;

}

func(0):

return 0;

5. (109)

func(8):

```
static int k=0;
if(n>0){//8>0 is TRUE
   k++; //static k is incremented to 1.
   return 2*func(n/2)+k;//func(4) is called.
   Returns (2*53+3) i.e 109
}
```

func(4):

```
static int k=0;
if(n>0){//4>0 is TRUE
   k++; //static k is incremented to 2.
   return 2*func(n/2)+k;//func(2) is called.
    Returns (2*25+3) i.e 53
}
```

func(2):

```
static int k=0;
if(n>0){//2>0 is TRUE
   k++; //static k is incremented to 3.
   return 2*func(n/2)+k;//func(1) is called.
   Returns (2*11+3) i.e 25
}
```

func(1):

```
static int k=0;
if(n>0){//1>0 is TRUE
   k++; //static k is incremented to 4.
   return 2*func(n/2)+k;//func(0) is called.
   Returns(2*4+3) i.e 11
}
```

f(0) returns (0+4). Static k is decremented to 3.

6. (b)

func(14, 1):

14%2 is 0, so else part is executed.

return func(7, 2)-1;// //Returns 14-1 i.e **13**.

func(7,2):

7%2 is 1, so else if part is executed.

return func(3, 4)+2; //Returns 12+2 i.e 14 to func(14, 1)

func(3,4):

3%2 is 1, so else if part is executed.

return func(1, 8)+4; //Returns 8+4 i.e 12 to func(7, 2)

func(1,8):

1%2 is 1, so else if part is executed.

return func(0, 16)+8; //fun(0,16) returns 0

//Returns 8 to func(3, 4)

7. (a)

func(7):

```
static int i=0;
if(n/2){//7/2= 3 is TRUE
   i--;//static i is decremented to -1
   return func(n/2)+i; //func(3) is called. func(7)
   returns -4-2 i.e -6
}else return i;
```

func(3):

```
static int i=0;
if(n/2){//3/2= 1 is TRUE
   i--;//static i is decremented to -2
   return func(n/2)+i; //func(1) is called. func(1)
   returns -2. func(3) returns -2-2 i.e -4
}else return i;
```

8. (c)

For i=0 in main():

display():

```
static int i;//i=0
if(i<=printf("GATE24")){//i<=6; printf() executed
   i=i+2;//i=2
   display();
}
```

display():

```
static int i;
if(i<=printf("GATE24")){//2<=6; printf() executed
   i=i+2;//i=4
   display();
}
```

display():

```
static int i;
if(i<=printf("GATE24")){//4<=6; printf() executed
   i=i+2;//i=6
   display();
}
```

```
display():
    static int i;
    if(i<=printf("GATE24")){//6<=6; printf() executed
        i=i+2;//i=8
        display();
    }

display():
    static int i;
    if(i<=printf("GATE24")){//8<=6 is FALSE but
        printf() executed
        i=i+2;
        display();
    }
```

For i=1 in main:

```
display():
    static int i;
    if(i<=printf("GATE24")){//8<=6 is FALSE but
        printf() executed
        i=i+2;
        display();
    }
```

For i=2 in main():

```
display():
    static int i;
    if(i<=printf("GATE24")){//8<=6 is FALSE but
        printf() executed
        i=i+2;
        display();
    }
```

Total number of times printf() executed is 7.