Programming in C

Recursion

DPP-06

[NAT]

1.
```
#include<stdio.h>
int func(int a){
    static int i=11;
    if(a<5) return i--;
    if(a>=5) {
        i=i+a;
        return func(a-1)+i;
    }
}
int main()
{
    printf("%d",func(9));
    return 0;
}
```
What is the value returned by func(9)?

_____

[NAT]

2.
```
#include<stdio.h>
int func(int a){
    static int i=3;
    if(a<3) return a+i;
    if(a>=3) {
        i=i*a--;
        return func(a-1)+func(a-2)-i;
    }
}
int main()

{
    printf("%d",func(7));
}
```
The value returned by func(7) is _____

[MCQ]

3.
```
#include<stdio.h>
int func(int a, int b){
    static int count=3;
    int c;
    if(a>=b){
        count=count&a;
        c=func(a-2,b-1)+a+b;
        printf("%d\t", c );
        return c;
    }
    if(a<b){
        count=count>>1;
        printf("%d\t",count);
        return a+b-count;
    }
}
```
The output printed by func(9, 7) is-

(a) 0 17 30 46 (b) 0 17 30 30

(c) 0 17 46 30 46 (d) 0 30 17 43

[NAT]

4.
```
#include<stdio.h>
int func(int a, int b){
    static int count=1;
    if(a<=b){
        count=count|b;
        return func(a+2, b+1)+count;
    }
    if(a>b){
        return count++;
    }
}
int main()
{
    printf("%d",func(1, 2));
    return 0;
}
```
The output is _____

[MCQ]

5.
```
#include<stdio.h>
void f1(int p){
    printf("%d\t", p);
    if(p<=3) f1(p+2);
    printf("%d\t",++p);
}
```

```
int main()
{
   f1(1);
   return 0;
}
```

The output printed is-

(a) 1 3 5 6 2 4 (b) 1 2 4 6 5 3
(c) 1 3 2 5 6 4 (d) 1 3 5 6 4 2

[MCQ]

6.
```
#include<stdio.h>
void output(int p){
   static int q=0;
   printf("%d", p);
   printf("%d", ++q);
   if(p>2) output(p-2);
   printf("%d", q++);
}
int main()
{
   output(4);
   return 0;
}
```

The output printed is-

(a) 421133 (b) 413332
(c) 412223 (d) 412222

[NAT]

7.
```
#include<stdio.h>
int f2(int a){
   static int b;
   return a+b--;
}
int f1(int a){
   static int b=1;
   b=f2(a);
   return a*b++;
}
int main(){
   int i, a=67, b=21;
   a+=f1(b)-f2(b);
   printf("%d",a);
   return 0;
}
```

The output printed is- _____

[MCQ]

8.
```
#include<stdio.h>
static int a=2;
void main(){
   extern int b;
   a++;
   printf("%d", a+b);
   static int a;
   a=5;
   a--;
   printf("%d", a+b);
}
int b = 10;
```

The output is-

(a) Garbage value
(b) Compilation error
(c) 13 12
(d) 13 14

[MCQ]

9. Which of the following storage classes is suited for loop variables?

(a) auto
(b) register
(c) static
(d) global

[MSQ]

10. Which of the following statements is/are TRUE?

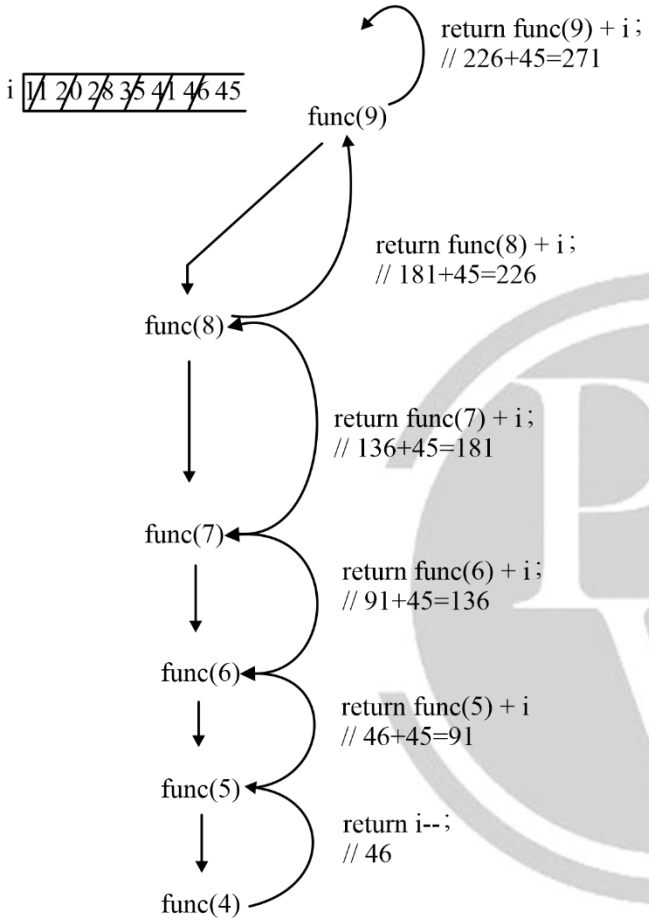
(a) Register variables may behave as auto variables.
(b) Staic variables can be declared and initialized once only.
(c) Static variables are stored in the heap segment.
(d) Auto variables contain garbage value if not initialized.

Answer Key

1. **(271)**
2. **(321)**
3. **(a)**
4. **(11)**
5. **(d)**
6. **(c)**
7. **(488)**
8. **(d)**
9. **(b)**
10. **(a, b, d)**

Hints and solutions

1. (271)



2. (321)

321

func(7) a 7 6 i = i * a --; = 3 * 7 = 21 return func (5) + fun (4) – i ⇒ 318 + 1263 – 1260 ⇒ 321	i 3 21 105 315 1260
func(5) a 5 4 i = i * a -- ; = 21 * 5 = 105 return func (3) + fun (2) – i ⇒ 316 + 317 – 315 ⇒ 318	
func(3) a 3 2 i = i * a -- ; = 105 * 3 = 315 return func(1) + fun(0) – 315; ⇒ 316 + 315 – 315 = 316;	
func(1) a 1 return a + i; ⇒ 1 + 315 = 316	
func(2) return a + i; // 2 + 315 = 317;	
func(4) a 4 3 i = i * a --; = 315 * 4 = 1260 return func(2) + func(1) – i; ⇒ 1262 + 1261 – 1260 = 1263	
func(2) a 2 return 1260 + 2;	
func(1) a 1 return 1 + 1260;	

3. (a)

func(9, 7) count 3 1 1 1

a 9 b 7

count = count & a
= 3 & 9 = 1
printf ("%d\t", func(7, 6) + a + b);
30 + 16 = 46

func(7, 6)

a 7 b 6

count = count & a
= 1 & 7 = 1
printf("d%\t", func(5, 5) + a + b);
17 + 13 = 30;

func(5, 5)

a 5 b 5

count = count & a
= 1 & 5 = 1
printf("%d\t", func(3, 4) + a + b);
7 + 10 = 17

func(3, 4)

a 3 b 4

count = count + >> 1
= 1 >> 1 = 0
printf("%d\t", count;// 0
return a + b – count;// 7
Output: 0 17 30 46

4. (11)

```
main()
printf ("%d", func(1,2)); //11
func(1, 2)
a 1   b 2   count 1 3 4
count = count | b;
      = 3
return func(3, 3) + count; // 7 + 4 = 11.
func(3, 3)
a 3   b 3
count = count | b;
      = 3
return func(5, 4) + count; // 3 + 4 = 7.
func(5, 4)
return count ++;  // return 3
```

∴ Output: 11

5. (d)

```
main( )
f1(1);
-----------------
f1(1)
 p 1
1. printf("%d\t", p); //1
2. (1<=3)→True
   f1(3)
3. printf("%d\t", ++p); //2
-----------------
f1(3)
 p 3
1. printf("%d\t", p); //3
2. (3<=3)→True
   f1(5)
3. printf("%d\t",++p); //4
-----------------
f1(5)
 P 5 6
1. printf("%d\t", p);//5
2. (5<=3)→ False
3. printf("%d\t",++p);//6
```

Output: 1 3 5 6 4 2

6. (c)

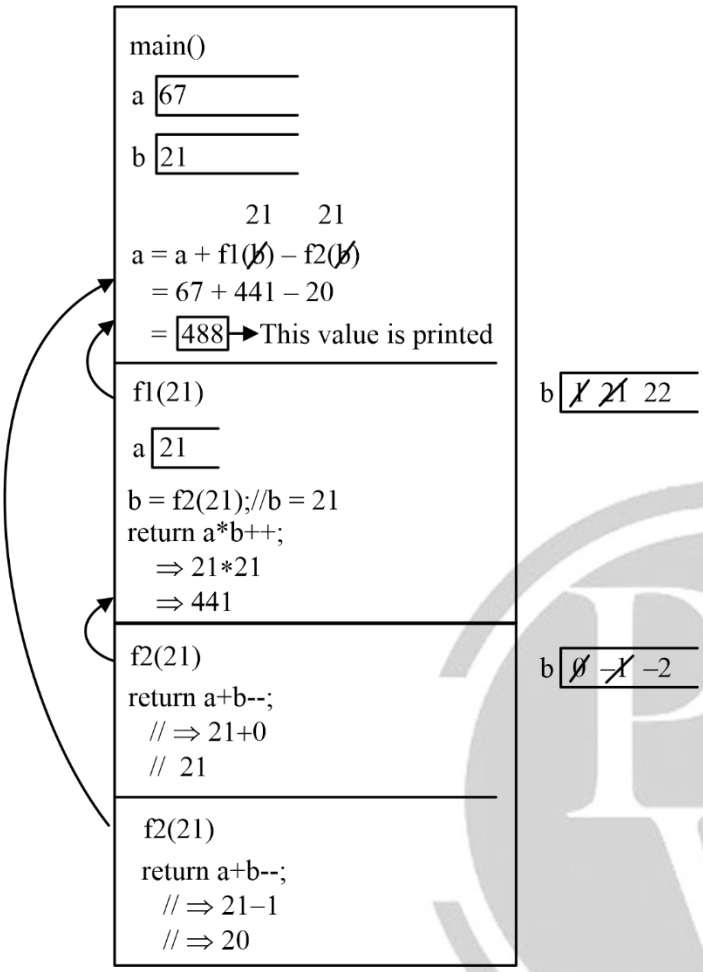
```
output(4)
 p 4
1. printf("%d", p); //4
2. printf("%d",++q); //1
3. if(p > 2)
      4 > 2 → True
      output (2);
4. printf("%d",q++);//3
-----------------
output(2)
 p 2
1. printf("%d", p); //2
2. printf("%d",++q);//2
3. 2 > 2 → False
4. printf("%d",q++);//2
```

q 0 1 2 3 4

Output:-

4 1 2 2 2 3

7. (488)



8. (d)

```
main()
1. extern int b:
2. a++;//Global var is incremented
3. printf("%d", a + b);//13
4. static int a;
5. a = 5
6. a--;//local var is decremented
7. printf("%d", a + b);//14
```

Global static
a 2 3
b 10
a 5 4
Local static

Output: 13 14

9. (b)

Register variables are the fastest so, they are best suited for loop variables.

10. (a, b, d)

(a) TRUE. Register variables may behave as auto variables.
(b) TRUE. Staic variables can be declared and initialized once only.
(c) FALSE. Static variables are stored in the stack segment
(d) TRUEAuto variables contain garbage value if not initialized.