C Programming

How many times printf statement will execute from Q1 to Q10

Q.1
```
for( i=1; i<=n; i=i*2)
    printf("pankaj");
```
→ $\lfloor \log_2 n \rfloor + 1$

Q.2
```
for( i= 1; i<=n; i=i*3)
   printf("GATE ACADEMY");
```
→ $\lfloor \log_3 n \rfloor + 1$

Q.3
```
for( i=n; i>=1; i=i/2)
   printf("Pankaj");
```
→ $\lfloor \log_2 n \rfloor + 1$

Q.4
```
for( i=n; i>=1; i=i/3)
   printf("GATE ACADEMY");
```
→ $\lfloor \log_3 n \rfloor + 1$

Q.5
```
for( i=1; i<=n; i++)
  {
  for(j=1; j<=10; j++)
     {
     printf("isko kahte hain nested loop");
     }
  }
```
→ $n \times 10 = 10n$

Q.6
```
for( i=1; i<=n; i++)
   {
   for( j=1; j<=n; j++)
     {
     printf("ye hai independent nested loop");
     }
   }
```
→ $n \times n = n^2$

Q.7
```
for( i=1; i<=n; i++)
   {
   for( j=1; j<=n; j=j*2)
```
→ $n \times (\lfloor \log_2 n \rfloor + 1)$

```
    {
      printf("ye hai independent nested loop");
    }
  }
```

Q.8
```
for( i=1; i<=n; i=i*2)
  {
   for( j=1; j<=n; j=j*2)
    {
     printf("ye hai independent nested loop");
    }
  }
```
→ $\left(\lfloor \log_2 n \rfloor + 1\right)^2$

Q.9
```
for( i=1; i<=n; i++)
  {
   for( j=1; j<=i; j++)
    {
     printf("ye hai dependent nested loop");
    }
  }
```
→ $\sum_{i=1}^{n} \sum_{j=1}^{i} (1) = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Q.10
```
for( i=1; i<=n; i++)
  {
   for( j=i; j<=3*i; j++)
    {
     printf("solve it using summation method");
    }
  }
```
→ $\sum_{i=1}^{n} \sum_{j=i}^{2i+1} (1) = \sum_{i=1}^{n} (i+1)+1 = \frac{n(n+1)}{2} + 2n$

Q.11 Consider the code :

```
void main ( ) {
int a [4] = {10, 20, 30, 40, 50}
printf ("%u ", a);
printf("%u ",a[0]);
printf ("%u",a+1);
printf ("%u ",& a + 1);
}
```

16 B

→ 1000

→ 10

→ 1004

→ 1000 + 16 = 1016

Assuming the base address of the array to be 1000 and size of integer is 4 byte, what are the values printed by printf( ) statements?

(A) 10 10 20 20

(B) 1000 1000 20 20

(C) 1000 10 20 1016

(D) 1000 10 1004 1016

Q.12 Consider the code:

```
void main ( ) {
    int a [2][3] = {1, 2, 3, 4, 5, 6};
    printf ("%u ", a);
    printf("%u ", a[0]);
    printf("%u", a[0]+1);
    printf ("%u",& a+1);
    printf("%u ", a[0][0]);
}
```

Assuming the base address of the array to be 1000 and size of integer is 4 byte, what are the values printed by printf( ) statements?

(A) 1000 1000 1004 1024 1

(B) 1000 1000 1012 1024 1

(C) 1000 1 1012 1024 1

(D) 1000 1000 1006 1024 1

Q.13

```
void main ( )
{
    int a[2][3][2] = {1, 2, 3, 4, 5, 6, 7,8,9,10,11,12};
    printf ("%u ", a);
    printf ("%u ", a[0]);
    printf ("%u ", a[0][0]);
    printf ("%u ",& a);
    printf ("%u ", a [0] + 1);
    printf ("%u ", a[0][0] +1);
}
```

Assuming the base address of the array to be 1000 and size of integer is 4 bytes, what are the values printed by printf( ) statements?

(A) 1000 1000 1 1000 1012 1004

(B) 1000 1000 1000 1000 1008 1004

(C) 1000 1000 1000 1000 1004 1002

(D) None of these

Q.14 What is the output of the following C code? Assume that the address of x is 2000 (in decimal) and an integer requires four bytes of memory.

```
#include <stdio.h>
int main()
```

```
{
unsigned int x[4][3] = {{1, 2, 3}, {4, 5, 6},
                        {7, 8, 9}, {10, 11, 12}};
 printf("%u, %u, %u", x+3, *(x+3), *(x+2)+3);
}
```

(A) 2036,2036,2036

(B) 2012,4,2204

(C) 2036,10,10

(D) 2012,4,6

* q ⇒ $x+3 = 2000 + 3 \times 12$

$= 2036$

* $(x+3)$ $\Delta x[3][0]$

$= 2000 + 12 \times 3$

$= 2036$

* $(x+2)+3 = \Delta x[2][0] + 3 \times 4$

$= (2000 + 12 \times 2) + 12$

$= 2024 + 12$

$= 2036$

Q.15 void main ( )

```
{
  int a [2][3] = {1, 2, 3, 4, 5, 6};
  printf ("%u %u %u ", a,*a,**a);
  printf ("%u %u %u %u ", a + 1,*a + 1,**a + 1);
}
```

1000, 1000, 1
1012, 1004, 2

Assuming the base address of the array to be 1000 and integer size to be 4 byte, what are the values printed by printf statements?

(A) 1000 1000 1 1012 1004 5
(B) 1000 1000 1000 1012 1004 1004
(C) 1000 1000 1 1012 1008 5
(D) None of these

Q.16 void main ( )

```
{
  int a[2][3][2] = {1, 2, 3, 4, 5, 6, 7,8,9,10,11,12};
  printf ("%u %u %u %u ", a,*a,**a,*** a);
  printf ("%u %u %u %u ", a + 1,*a + 1,**a + 1,*** a + 1);
}
```

1000, 1000, 1000, 1
1024, 1008, 1009, 2

Assuming the base address of the array to be 1000 and the size of integer is 4 bytes, what are the values printed by printf statements.

Q.17 What does the following declaration signifies :

(A) int (*p) [4] → p is a pointer to an array of 4 integers
(B) int * (*p) [5] → p is a pointer to an array of 5 pointer to integers
(C) int * p[10] → p is an array of 10 ~~integers~~ pointer to integer
(D) int (*p) ( ) → p is a pointer to fn that takes no args & it return an integer
(E) int (*p) (int, int) → p is pointer to fn that takes 2 integer args. & return an int
(F) int (*p) (char *a) → p is a pointer to fn that takes 1 pointer to char args & return integer

Q.18 What does the following C-statement declare? int (* f) (int *) ; **GATE -2005/ISRO-2017**

(A) A function that takes an integer pointer as argument and returns an integer.

(B) A function that takes an integer as argument and returns an integer pointer.

(C) A pointer to a function that takes an integer pointer as argument and returns an integer.

(D) A function that takes an integer pointer as argument and returns a function pointer.

Q.19 What will be the output when you will execute the following *C* code?

```
void main ( ) {
        int a[2][3] = {5,10,15, 20, 25, 30};
        int     (*p)[2][3]=&a;
        printf ("%d \t ",*** p);
        printf("%d\t",***(p+1));
        printf("%d\t",**(*p+1));
        printf("%d \t ",*(*(*p +1) + 2));
        }
```

(A) 5 Garbage 20 30

(B) 10 15 30 20

(C) 5 15 20 30

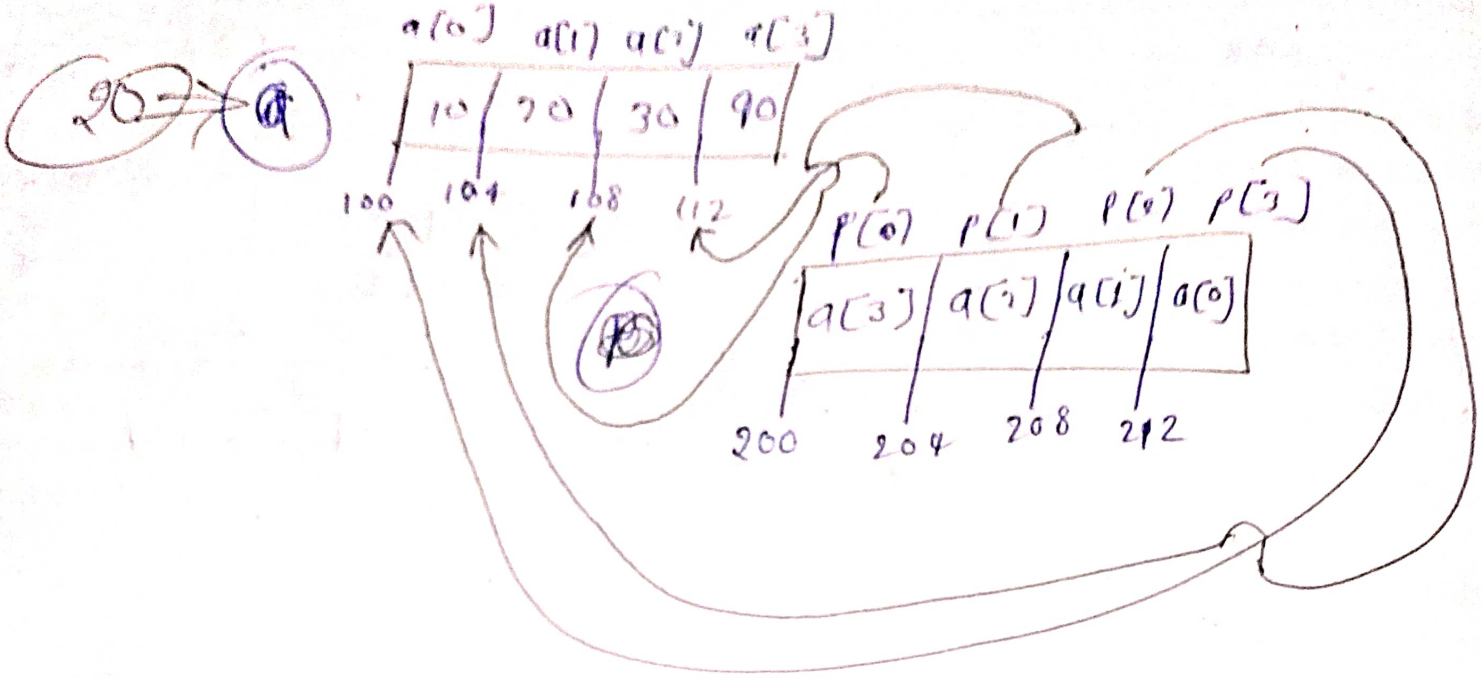
(D) C.E

(E) None of the above

Q.20

```
#include<stdio.h>
int main()
{
int a[4]={10,20,30,40};
int *p[4]={a+3,a+2,a+1,a};
int y;
y=--p[0]-p[1];
printf("%d",y);
printf("%d",*p[0]);
}
```

(A) 0 20 (B) 1 30 (C) 0 30 (D) 0 10



$P[0] = 112 \Rightarrow --P[0] = 108 \Rightarrow 30$

$P[1] = 108 = 30$

$y = 108 - 108 = 0$

$P[0] = 30$