Gate Overflow

1.

2.



3.

function(n)

{

for( i = 1 ; i <= n ; i + + )

for( j = 1 ; j <= n ; j+ = i )

print("\*") ;

}

4.

int j=0;

for(i=0;i<n;i++)

{

for(i=0;i<2n;i++)

{

while(j<n)

{

j++;

}

}

}

time complexity.?

a.O(n2)O(n2)                b.O(n4)O(n4)                       c.O(n3)O(n3)                    d.O(n)

5.

for(int i = 1; i<=n; i++)

{

//int i=n;

for(int j=i; j<i\*i; j++)

{

if(j%i==0)

{

for(int k =0; k<j; k++)

{

counter++;

}

}

}

}

6.

#include <stdio.h>

#define N 3

int main() {

int array[N] = {1,2,3};

int i,j;

for ( i=1; i<(1<<N); i++) {

for( j=0; j<N; j++) {

if((1<<j)&i) {

printf("%d", array[j]);

}

}

printf("\n");

}

return 0;

}

* A. How many times the if **successfully** executes in this instance of c program. And how many time when N=n,n is a positive integer N=n,n is a positive integer  ?
* B. What is the output?
* C. What will be the complexity when NN is large.

**7.**

for(i=1 to n)  
    {  
        if(n mod i==0)  
        {  
            for(int j=1 to n)  
            printf(j);  
        }  
    }

8.

**for(i=n;i>0;i/=2)**

**{**

**k=1;**

**for(j=1;j<=n;j=j+k)**

**k++;**

**}**

9.

sum=0;

for(i=0;i<=n;i++) {

for(j=1;j<=i;j++) {

if(j%i==0) {

for(k=0;k<=n;k++) {

sum=sum+k;

}

}

}

}

10.

int count = 0;

for (int i = N; i > 0; i /= 2) {

for (int j = 0; j < i; j++) {

count += 1;

}

}

11.

For(I=1 ; I<=n ; I++)

{

For(J=1 ; J<=I ; J++)

{

For(K=1 ; K<=n^5 ; K=15 × K)

{

x=y+z;

}

}

}

12.

void fun(int n, int k)

{

for (int i=1; i<=n; i++)

{

int p = pow(i, k);

 for (int j=1; j<=p; j++)

    {           // Some O(1) work

  }

}

}

13.

sum=0;

for(i=0;i<n;i++)

for(j=0;j<i\*i;j++)

for(k=0;k<j;k++)

sum++;

14.

while (j >= 1){      
  
 for (i = 1 to j    )     
  
   x = x + 1    ;  
  
 j = n/2 ;  
  
 }

15.

main()

{

for(i=1;i<=n;i++)

{

for(i=1;i<=n\*n;i++)

{

for(i=1;i<=n\*n\*n;i++)

{

x=y+z;

}

}

}

}

16.

for(int i=0; i<n; i++)

for(int j=i+1; j<n; j++)

for(int k=j+1; k<n; k++

17.

int IsPrime(n)

{

int i, n;

for (i=2; i<=sqrt(n);i++)

if(n%i == 0)

{printf("Not Prime \n"); return 0;}

return 1;

}