Question 1.

```
int j, n;
j=1;
while(j <= n)
 j = j * 2;     -------> [logn]+1
```

Question 2.

```
int f1(int n) {
   if (n ==  0  || n ==  1 ){
       return  n;
  }
   return  ( 2  * f1(n -  1 ) +  3  * f1(n -  2 ));
}
int f2(int n) {
   int  i;
   int  X[N], Y[N], Z[N];
  X[ 0 ] = Y[ 0 ] = Z[ 0 ] =  0 ;
  X[ 1 ] =  1 ; Y[ 1 ] =  2 ; Z[ 1 ] =  3 ;
   for (i =  2 ; i <= n; i++){
   X[i] = Y[i -  1 ] + Z[i -  2 ];
   Y[i] =  2  * X[i];
   Z[i] =  3  * X[i];
  }
   return  X[n];
}
```

The returning time of f1(n) and f2(n) are-------→@(2^n) and@(n)

Question 3.

```
int DoSomething(int n){
 if(n <= 2)
     return 1;
 else
     return (floor(sqrt(n)) + n);
}                    ------------→@(loglogn)
```

Question 4.

```
int IsPrime(n){
 int i, n;
 for(i=2; i<=sqrt(n);i++){
  if(n % i == 0){
    printf("No prime\n"); return 0;
  }
  return 1;
 }
}              ---------→O(n'1/2) and omega(1)
```

Question 5.

```
int gcd(n,m)
{
if (n % m == 0) return m;
n = n % m;
return gcd(m,n);
}                                                ----→@(logn)
```

Question 6.

```
double foo(int n){
 int i;
 double sum;
 if(n == 0) return 1.0;
 sum = 0.0;
 for (i = 0; i < n; i++){
  sum += foo(i);
 }
 return sum;
```

```
}                    ------→O(n)
```

Question 7.

```
counter = 0;
for(i = 1; i <= n; i++){
 if(A[i]==1){
   counter++;
 }else{
   f(counter); counter = 0;
 }
}                 -----→@(n)
```

Question 8.

```
int recursive(int n){
 if(n == 1){
   return (1);
 }
 return (recursive(n - 1) + recursive(n - 1));
}                   ------------→O(2'n)
```

Question 9.

```
x = m;
y = 1;
while(x - y > ε){
 x = (x + y) / 2;
 y = m/x;
}
print(x); -----------→m^1/2
```

Question 10.

```
int i, j, k=0;
for (i = n/2; i <= n; i++)
for (j = 2; j <= n; j = j*2)
k = k + n/2;
return (k); ---------------------→O(n^2 logn)
```

Question 11.

```
int fun(int n)
{
    int i, j;
    for (i = 1; i <= n ; i++)
    {
        for (j = 1;  j < n; j += i)
        {
            printf("%d %d", i, j);
        }
    }
}
```

----------→o(n log n)