1- Please calculate time complexities of the following code snippets using **Sigma Σ (Mathematical)** notations and present the final complexity using bigO notation.

Note: You can omit/disregard time complexities that would come from **for** loops. Please only calculate time complexities based on the code lines with **print** command. Final bigO complexity will not change.

Example:

```
for (int i = 1; i <= n; i++) {
      print(“hi”);      //    1
      print(“bye”);     //    1
}
```

$$\sum_{i=1}^{n} 2 = 2n$$

```
a: for (int i = 1; i <= 5; i++)
      for (int j = 0; j < 2*n; j++)
            print(“hi”);

b: for (int i = 1; i <= n; i++)
      for (int j = 0; j < n; j++)
            print(“hi”);

c: for (int i = 1; i <= n; i++) {
      print(“1.loop”);
      for (int j = i; j <= n; j++)
            print(“2.loop”);
   }

d: for (int i = 1; i <= n; i++) {
      print(“1.loop”);
      for (int j = 1; j <= i; j++)
            print(“2.loop”);
   }

e: for (int j = 1; j <= n; j*=2)
      print(“hi”);

f: for (int j = 2; j <= n; j=j^2)
      print(“hi”);

g: for (int i = 1; i <= n; i += 2)
      for (int j = 1; j <= n; j += 2)
         for (int k = 1; k <= n; k *= 2)
            print(“hi”);

h: for (int i = 1; i <= n; i += 2)
      for (int j = 1; j <= n; j += 2)
         for (int k = j; k <= n; k++)
            print(“hi”);

i: for (int i = 1; i <= n; i++)
      for (int j = 1; j <= n; j *= 2)
         for (int k = 1; k <= j; k++)
            print(“hi”);

j: for (int i = 1; i <= n; i += 2) {
      for (int j = 1; j <= i; j++) {
         for (int k = 1; k <= n; k += 5)
            print(“hi”);
         for (int k = 1; k <= n; k *= 2)
            print(“hi”);
      }
   }
```

2- Please write a **linear** function (in a regular language or pseudocode) that calculates **factorial** ! of a given number, calculate the time complexity of the function and present the final complexity using bigO notation.

3- Please write a **recursive** function (in a regular language or pseudocode) that calculates **factorial** ! of a given number, calculate the time complexity of the function and present the final complexity using bigO notation.

4- Please write a **linear** function (in a regular language or pseudocode) that calculates **fibonacci** value of a given index number, calculate the time complexity of the function and present the final complexity using bigO notation.

5- Please write a **recursive** function (in a regular language or pseudocode) that calculates **fibonacci** value of a given index number, calculate the time complexity of the function and present the final complexity using bigO notation.