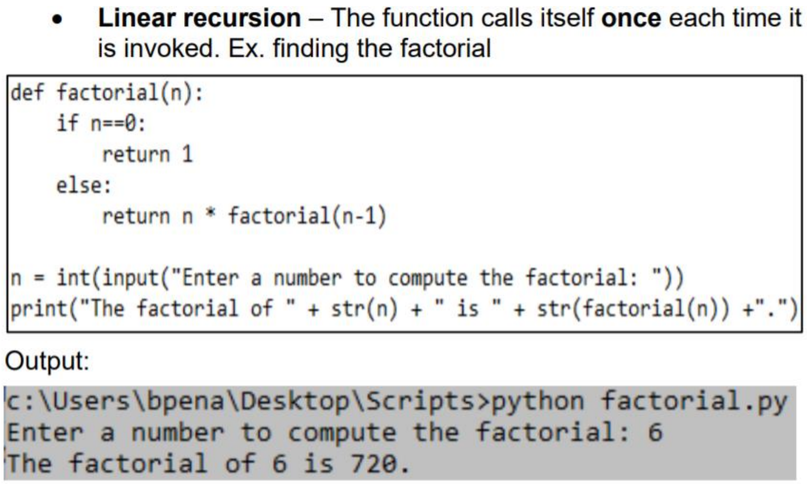
TASK PERFORMANCE 1 MIDTERM

ABLERO, ANGELICA A.

1. **Convert one sample Python script in 02 Handout 1 into Java code.**

Linear recursion

Python script



- **Linear recursion** – The function calls itself **once** each time it is invoked. Ex. finding the factorial

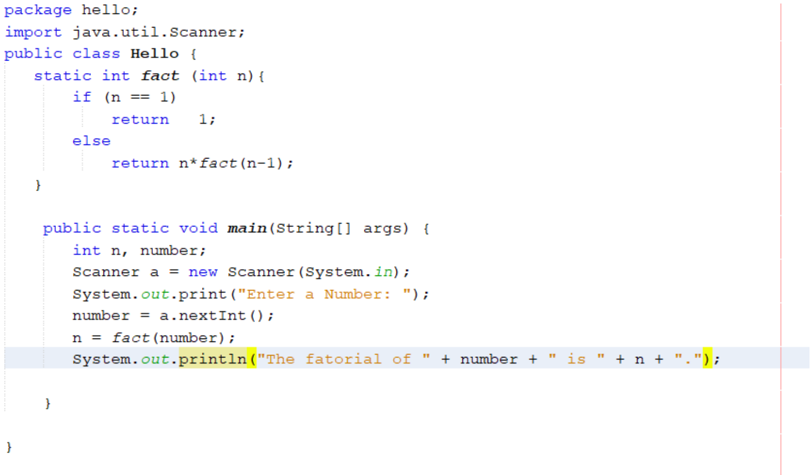
```
def factorial(n):
    if n==0:
        return 1
    else:
        return n * factorial(n-1)

n = int(input("Enter a number to compute the factorial: "))
print("The factorial of " + str(n) + " is " + str(factorial(n)) +".")
```

Output:

```
c:\Users\bpena\Desktop\Scripts>python factorial.py
Enter a number to compute the factorial: 6
The factorial of 6 is 720.
```

Syntax:



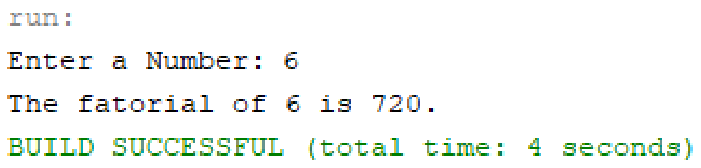
```
package hello;
import java.util.Scanner;
public class Hello {
   static int fact (int n){
       if (n == 1)
           return   1;
       else
           return n*fact(n-1);
   }

    public static void main(String[] args) {
       int n, number;
       Scanner a = new Scanner(System.in);
       System.out.print("Enter a Number: ");
       number = a.nextInt();
       n = fact(number);
       System.out.println("The fatorial of " + number + " is " + n + ".");

    }

}
```

Output:

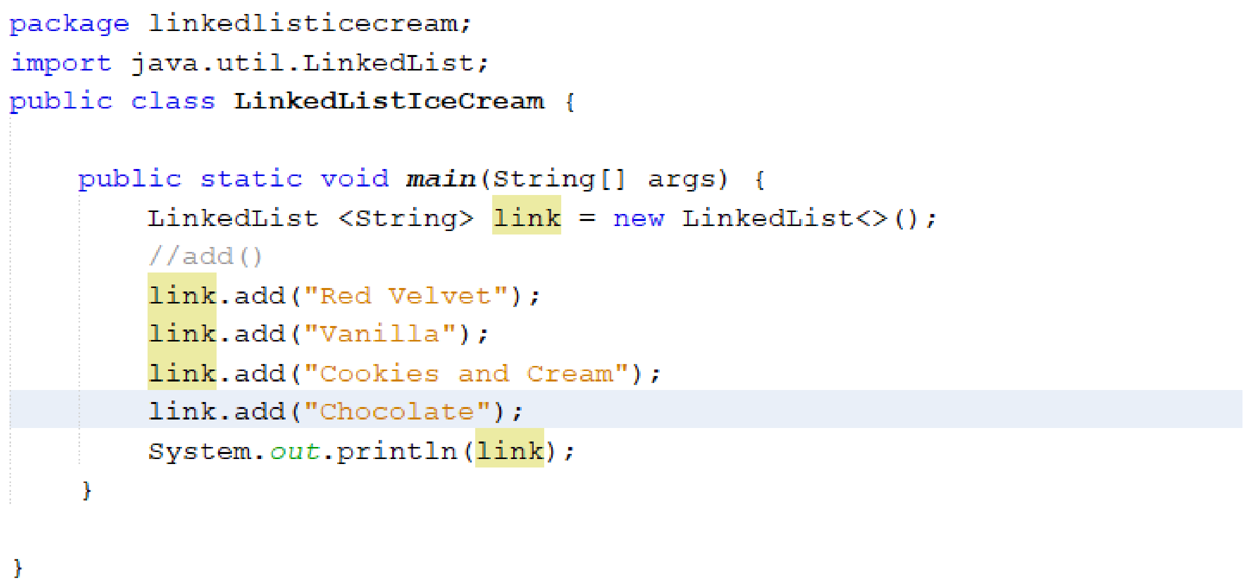


```
run:
Enter a Number: 6
The fatorial of 6 is 720.
BUILD SUCCESSFUL (total time: 4 seconds)
```

2. **Create and print a linked list with four elements.**

Ice Cream Flavors

SYNTAX:

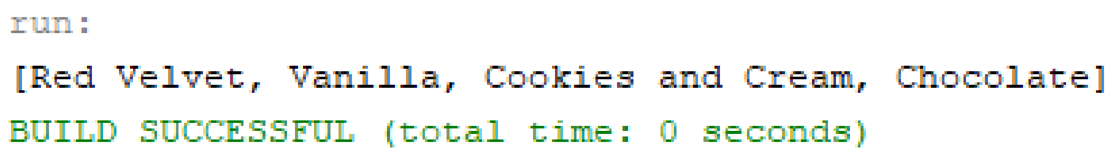


```
package linkedlisticecream;
import java.util.LinkedList;
public class LinkedListIceCream {

    public static void main(String[] args) {
        LinkedList <String> link = new LinkedList<>();
        //add()
        link.add("Red Velvet");
        link.add("Vanilla");
        link.add("Cookies and Cream");
        link.add("Chocolate");
        System.out.println(link);
    }

}
```

OUTPUT:



```
run:
[Red Velvet, Vanilla, Cookies and Cream, Chocolate]
BUILD SUCCESSFUL (total time: 0 seconds)
```