PRACTICAL-1

Given a string and a non-negative int n, we'll say that the front of the string is the first 3 chars, or whatever is there if the string is less than length 3. Return n copies of the front; front_times('Chocolate', 2) → 'ChoCho' front_times('Chocolate', 3) → 'ChoChoCho' front_times('Abc', 3) → 'AbcAbcAbc'

Program Code:

```
class SP_6
{
	public static void main(String args[])
	{
		SP_6 obj=new SP_6();
		String s1=obj.front_lines("Chocolate",2);
		String s2=obj.front_lines("Chocolate",3);
		String s3=obj.front_lines("ABC",3);
		System.out.println(s1);
		System.out.println(s2);
System.out.println(s3);

	}
	 String front_lines(String s, int no)
	{
		String s1="";
for(int i=1;i<=no;i++)
		{
			s1=s1+s.substring(0,3);
		}
		return s1;
```

```
        }
}
```

Output:

```
C:\Java\JAVA_practicals>javac SP_6.java

C:\Java\JAVA_practicals>java SP_6
ChoCho
ChoChoCho
ABCABCABC
```

PRACTICAL-2

Given an array of ints, return the number of 9's in the array.

array_count9([1, 2, 9]) → 1 array_count9([1, 9, 9]) → 2 array_count9([1, 9, 9, 3, 9]) → 3

Program Code:

```
class SP_7
{
	public static void main(String[] args)
	{
		SP_7 obj=new SP_7();
	int x1[]={1,2,9};		int x2[]={1,9,9};		int x3[]={1,9,9,3,9};		int n1=obj.array_count(x1);		int n2=obj.array_count(x2);		int n3=obj.array_count(x3);
System.out.println(n1);
		System.out.println(n2);
System.out.println(n3);

	}
	int array_count(int[] arr)
	{
		int cnt=0;
		for(int i=0;i<arr.length;i++)
		{
```

```
                    if(arr[i]==9)
                        cnt++;
                }
                return cnt;
        }
}
```

OUTPUT:

```
C:\Java\JAVA_practicals>java SP_7.java
1
2
3
```

PRACTICAL-3

Given an array of ints, return True if one of the first 4 elements in the array is a 9.

The array length may be less than 4.

array_front9([1, 2, 9, 3, 4]) → True array_front9([1, 2, 3, 4, 9]) → False array_front9([1, 2, 3, 4, 5]) → False

PROGRAM CODE:

```
class SP_8
{
	public static void main(String[] args)
	{
		int[] a1={1,2,9,3,4};
int[] a2={1,2,3,4,9};		int[]
a3={1,2,3,4,5};			SP_8 s1=new
SP_8();			boolean
b1=s1.array_front9(a1);			boolean
b2=s1.array_front9(a2);			boolean
b3=s1.array_front9(a3);
		System.out.println(b1);
		System.out.println(b2);
System.out.println(b3);

	}
	boolean array_front9(int[] arr)
	{
		int n=arr.length;
```

```
            if(n>4)
                n=4;
            for(int i=0;i<n;i++)
            {
                if(arr[i]==9)
            return true;
            }
                return false;

    }
}
```

OUTPUT:

```
C:\Java\JAVA_practicals>java SP_8.java
true
false
false
```

PRACTICAL-4

Given a string, return a string where for every char in the original, there are two chars.

double_char('The') → 'TThhee' double_char('AAbb') → 'AAAAbbbb' double_char('Hi-There') → 'HHii--TThheerree'

PROGRAM CODE:

```
class SP_9
{
	public static void main(String[] args)
	{
		SP_9 s=new SP_9();
		String s1=s.double_char("ABC");
		System.out.println(s1);
	}
	String double_char(String str)
	{
		String str2="";
		for(int i=0;i<str.length();i++)
		{
			str2=str2+str.charAt(i)+str.charAt(i);
		}
		return str2;
	}
}
```

OUTPUT:

```
C:\Java\JAVA_practicals>javac SP_9.java

C:\Java\JAVA_practicals>java SP_9
AABBCC
```

PRACTICAL-5

Write a program that will reverse the sequence of letters in each word of your chosen paragraph. For instance, "To be or not to be" would become "oT e bro ton ot eb".

Practical code:

```
import java.util.*; class
SP_10
{
        public static void main(String[] args)
        {
                String tmp;
                StringTokenizer s=new StringTokenizer("To be or not to be");
while(s.hasMoreTokens())
                {
                        tmp=s.nextToken();
                        for(int i=tmp.length();i>0;i--)
                        {
                                System.out.print(tmp.charAt(i-1));
                        }
                        System.out.print("  ");
                }
        }
}
```

OUTPUT:

```
C:\Java\JAVA_practicals>java SP_10.java
oT  eb  ro  ton  ot  eb
```