Assignment 02

Task 01:

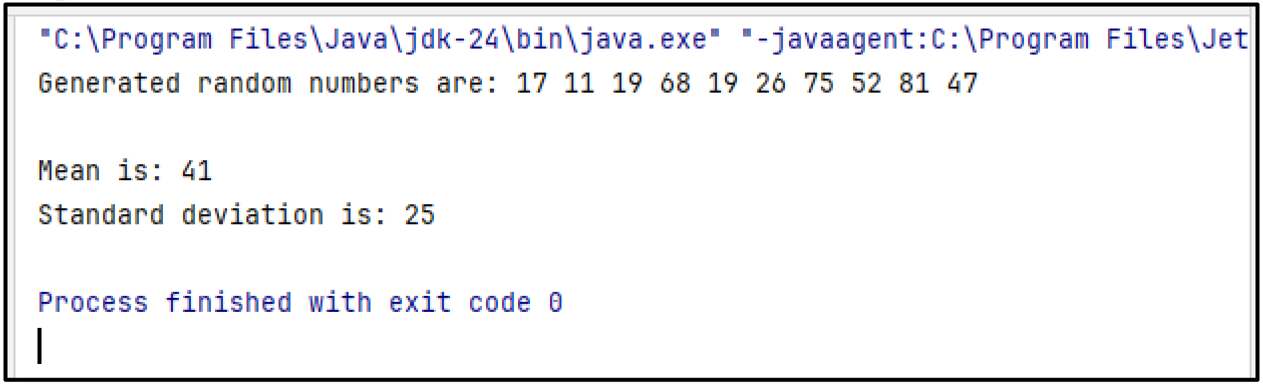
Generate 10 random numbers and compute the mean and standard deviation.

Code:

```
package Task01;
import java.lang.Math;

public class MeanStdDeviation {

    // method to calculate mean
    public int calMean(int[] nums){
        int mean = 0;

        for(int n : nums){
            mean += n;
        }
        mean /= nums.length;

        return mean;
    }

    // method to calculate variance
    public int calVariance(int[] nums){
        int mean = calMean(nums);
        int variance = 0;

        for(int num : nums) {
            variance += ((num - mean) * (num - mean));
        }
        variance /= nums.length;

        return variance;
    }

    // method to calculate standard deviation
    public int calStdDeviation(int[] nums){
        int var = calVariance(nums);

        int sd = (int)Math.sqrt(var); // typecasting into int, because sqrt()
gives ans in double

        return sd;
    }
}
```

```
package Task01;
import java.util.Random;

public class Main {

    // method to generate 10 random numbers
    public static int[] generateRandom(){
        Random rand = new Random();
        int[] nums = new int[10];

        // generating random nums
//        for(int n : nums){
//            n = rand.nextInt(100);
//        }

        for(int i = 0; i < nums.length; i++){
            nums[i] = rand.nextInt(100);
        }

        return nums;
    }

    public static void main(String[] args) {
        MeanStdDeviation obj = new MeanStdDeviation();
        int[] nums = generateRandom();

        System.out.print("Generated random numbers are: ");
        for(int n : nums){
            System.out.print(n + " ");
        }

        System.out.println("\n\nMean is: " + obj.calMean(nums));
        System.out.println("Standard deviation is: " +
obj.calStdDeviation(nums));

    }
}
```

Output:



```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\Jet
Generated random numbers are: 17 11 19 68 19 26 75 52 81 47

Mean is: 41
Standard deviation is: 25

Process finished with exit code 0
```

Task 02:

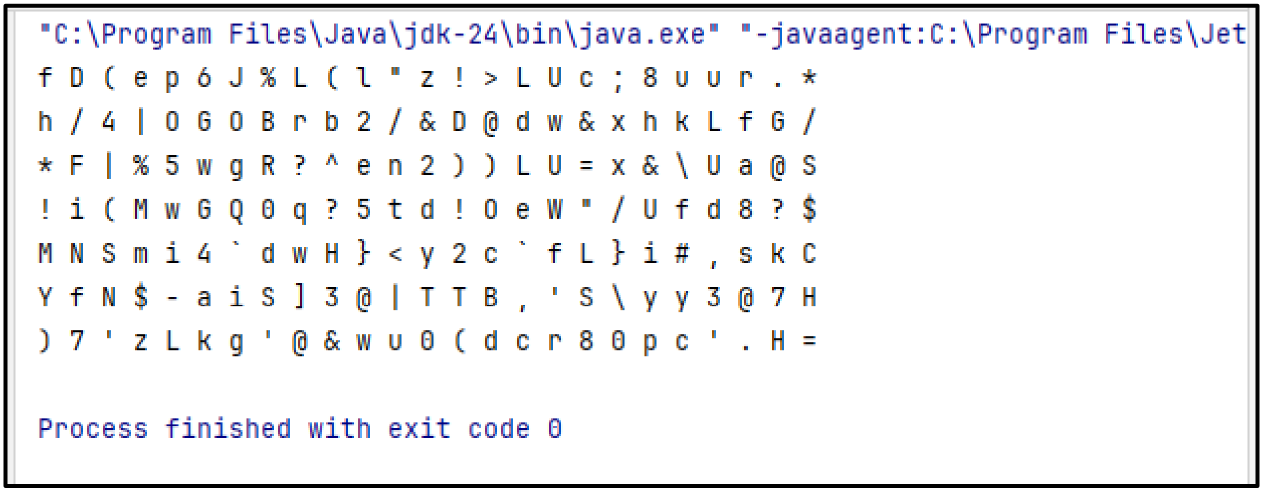
Write the methods for generating random characters. The program uses these methods to generate 175 random characters between ‘!' and ‘~' and displays 25 characters per line.

To find out the characters between ‘!' and ‘~', see Appendix B, “The ASCII Character Set.”

Code:

```
package Task02;
import java.util.Random;

public class RandomCharacter {

    // method to generate random numbers (ascii code)
    public static int[] generateRandom(){
        Random rand = new Random();

        int[] nums = new int[175];

        // generating 175 random ascii codes
        for(int i = 0; i < nums.length; i++){
            nums[i] = rand.nextInt(33, 126);
        }

        return nums;
    }

    public static void main(String[] args) {
        int[] nums = generateRandom();

        // typecasting them into chars
        char[] chars = new char[175];
        for(int i = 0; i < 175; i++){
            chars[i] = (char)nums[i];
        }

        // displaying them
        for(int i = 0; i < chars.length; i++){
            System.out.print(chars[i] + " ");

            if((i+1) % 25 == 0){
                System.out.println();
            }
        }

    }
}
```

Output:



```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\Jet
f D ( e p 6 J % L ( l " z ! > L U c ; 8 u u r . *
h / 4 | O G O B r b 2 / & D @ d w & x h k L f G /
* F | % 5 w g R ? ^ e n 2 ) ) L U = x & \ U a @ S
! i ( M w G Q 0 q ? 5 t d ! O e W " / U f d 8 ? $
M N S m i 4 ` d w H } < y 2 c ` f L } i # , s k C
Y f N $ - a i S ] 3 @ | T T B , ' S \ y y 3 @ 7 H
) 7 ' z L k g ' @ & w u 0 ( d c r 8 0 p c ' . H =

Process finished with exit code 0
```

Task 03:

Write a method to generate Fibonacci sequence.

0 1 1 2 3 5 8 13 21 34 55 89…

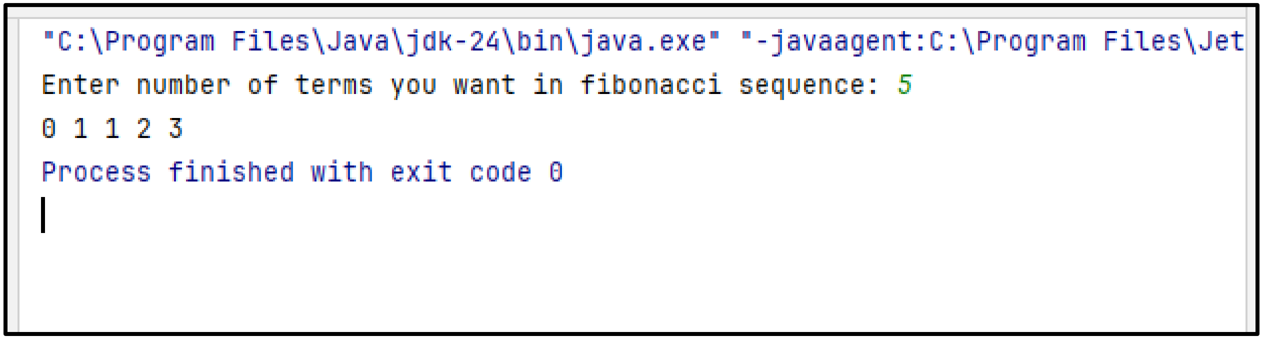
Hint:

fib(2) = fib(0) + fib(1);
fib(0) = 0;
fib(1) = 1;
fib(n) = fib(n-2) + fib(n-1); n>=2

Code:

```java
package Task03;
import java.util.Scanner;

public class Fibonacci {

    // method to calculate fibonacci number
    public static int fib(int n){
        if(n == 0 || n == 1){
            return n;
        }
        return fib(n-2) + fib(n-1);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of terms you want in fibonacci
sequence: ");
        int term = sc.nextInt();

        for(int i = 0; i < term; i++){
            System.out.print(fib(i) + " ");
        }

        sc.close();
    }
}
```

Output:



```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\Jet
Enter number of terms you want in fibonacci sequence: 5
0 1 1 2 3
Process finished with exit code 0
```

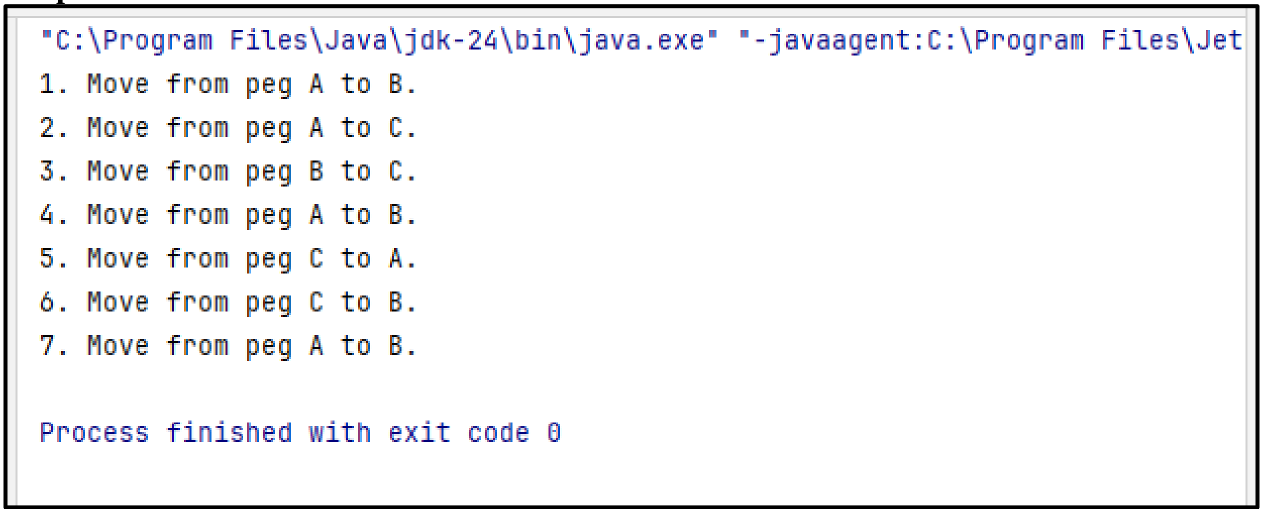
Task 04:

Write an algorithm to solve tower of Hanoi problem.

Code:

```
package Task04;

public class TowerOfHanoi {

    // integer for numbering
    private int sNo = 0;

    // method for tower of hanoi
    public void toh(char a, char b, char c, int n){
        // base case
        if(n == 1){
            System.out.println(++sNo + ". Move from peg " + a + " to " + b +
".");
        }

        // recursive case
        else{
            toh(a, c, b, n-1);
            System.out.println(++sNo + ". Move from peg " + a + " to " + b +
".");
            toh(c, b, a, n-1);
        }
    }

    public static void main(String[] args) {
        TowerOfHanoi obj = new TowerOfHanoi();

        obj.toh('A', 'B', 'C', 3);
    }

}
```

Output:



```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\Jet
1. Move from peg A to B.
2. Move from peg A to C.
3. Move from peg B to C.
4. Move from peg A to B.
5. Move from peg C to A.
6. Move from peg C to B.
7. Move from peg A to B.

Process finished with exit code 0
```