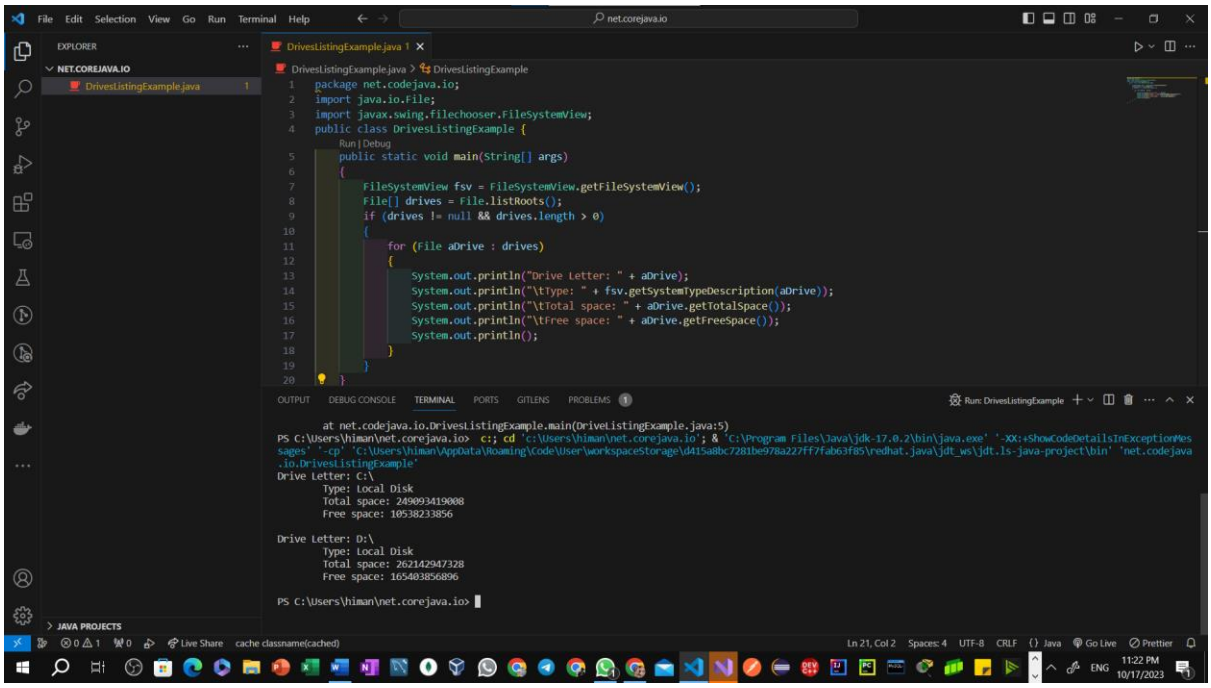
Q1- Write a program that lists all available disk Partitions with their drive letters, type descriptions, free spaces and total spaces.



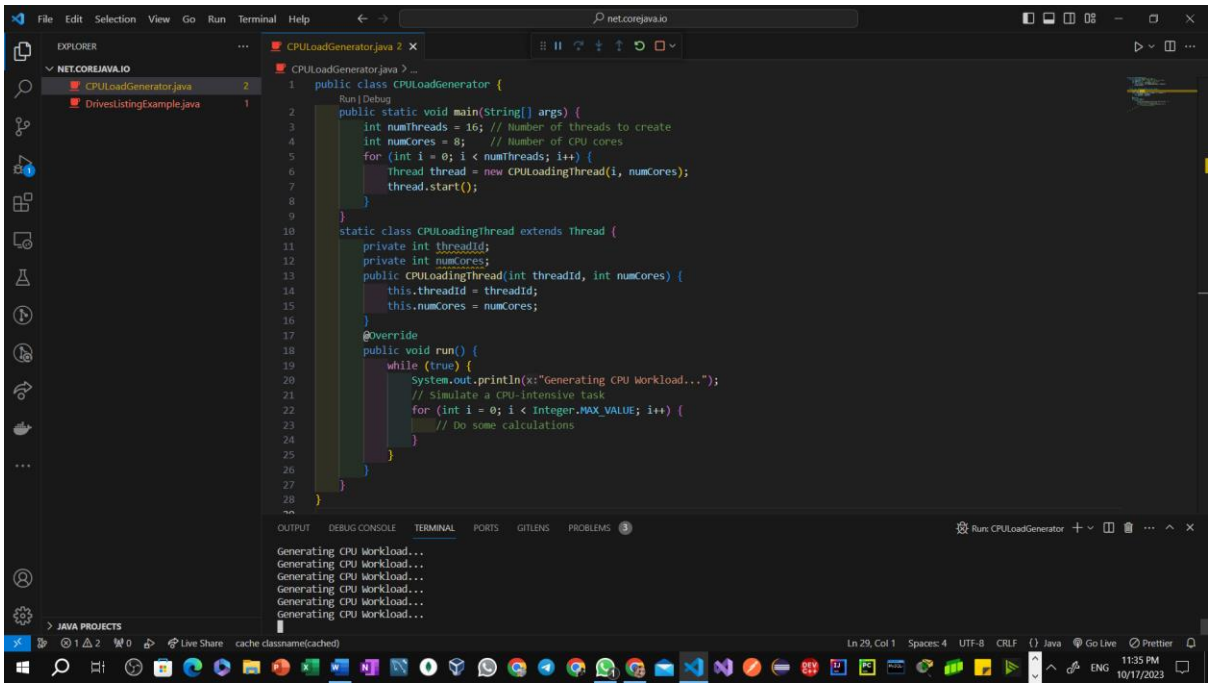
```
package net.codejava.io;
import java.io.File;
import javax.swing.filechooser.FileSystemView;
public class DrivesListingExample {
    public static void main(String[] args)
    {
        FileSystemView fsv = FileSystemView.getFileSystemView();
        File[] drives = File.listRoots();
        if (drives != null && drives.length > 0)
        {
            for (File aDrive : drives)
            {
                System.out.println("Drive Letter: " + aDrive);
                System.out.println("\tType: " + fsv.getSystemTypeDescription(aDrive));
                System.out.println("\tTotal space: " + aDrive.getTotalSpace());
                System.out.println("\tFree space: " + aDrive.getFreeSpace());
                System.out.println();
            }
        }
    }
}
```

```
        at net.codejava.io.DrivesListingExample.main(DriveListingExample.java:5)
PS C:\Users\himan\net.corejava.io>  c:; cd 'c:\Users\himan\net.corejava.io'; & 'C:\Program Files\Java\jdk-17.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMes
sages' '-cp' 'C:\Users\himan\AppData\Roaming\Code\User\workspaceStorage\d415a8bc7281be978a227ff7fab63f85\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'net.codejava
.io.DrivesListingExample'
Drive Letter: C:\
        Type: Local Disk
        Total space: 249093419008
        Free space: 10538233856

Drive Letter: D:\
        Type: Local Disk
        Total space: 262142947328
        Free space: 165403856896

PS C:\Users\himan\net.corejava.io>
```

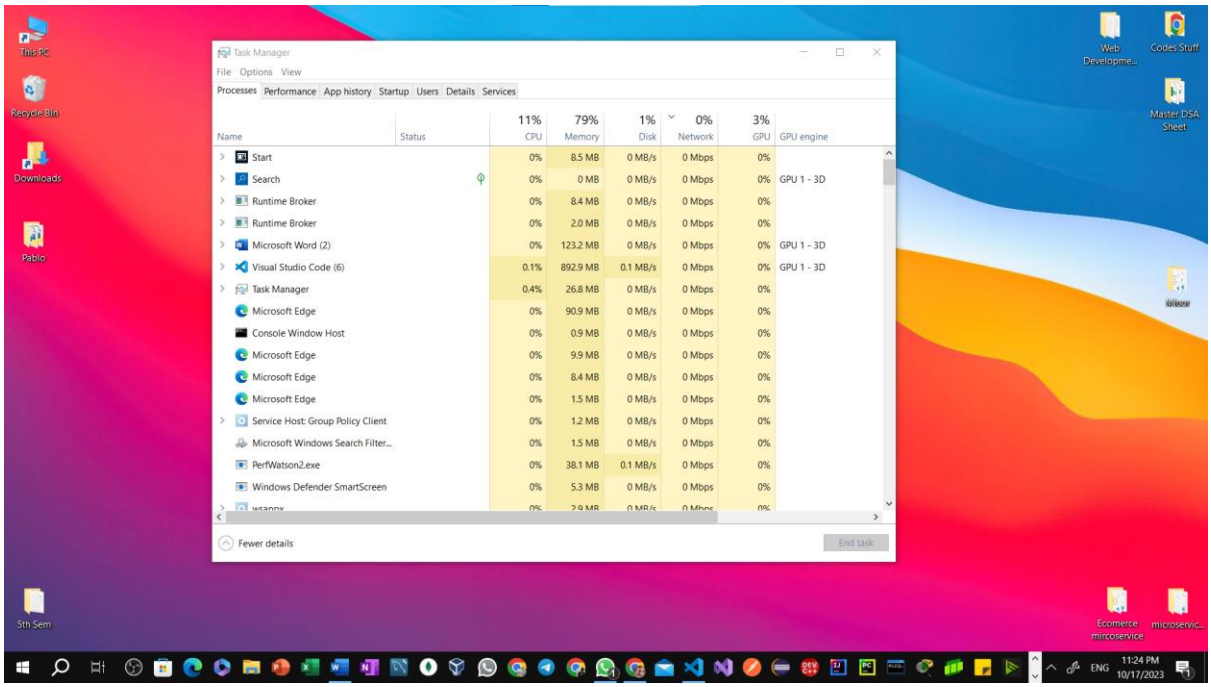
Q2- Write a program in java to generate CPU load (CPU usage upto 100%)



```
public class CPULoadGenerator {
    public static void main(String[] args) {
        int numThreads = 16; // Number of threads to create
        int numCores = 8;    // Number of CPU cores
        for (int i = 0; i < numThreads; i++) {
            Thread thread = new CPULoadingThread(i, numCores);
            thread.start();
        }
    }
    static class CPULoadingThread extends Thread {
        private int threadId;
        private int numCores;
        public CPULoadingThread(int threadId, int numCores) {
            this.threadId = threadId;
            this.numCores = numCores;
        }
        @Override
        public void run() {
            while (true) {
                System.out.println(x:"Generating CPU Workload...");
                // Simulate a CPU-intensive task
                for (int i = 0; i < Integer.MAX_VALUE; i++) {
                    // Do some calculations
                }
            }
        }
    }
}
```

```
Generating CPU Workload...
Generating CPU Workload...
Generating CPU Workload...
Generating CPU Workload...
Generating CPU Workload...
Generating CPU Workload...
```

Output without running (CPU utilization=11%)



Output when running (CPU utilization=100%)