LINEAR PROGRAMMING

import java.io.\*;

import java.util.Scanner;

public class LP

{

public static void main (String args[]) throws IOException

{

BufferedReader inpt = new BufferedReader(new InputStreamReader(System.in));

int[] cost = new int[3];

int[] availability = new int[3];

int[] storage = new int[3];

String[] cloud\_p = new String[3];

for (int i = 0; i < cloud\_p.length; i++) {

System.out.print("Enter Provider name: ");

cloud\_p[i] = inpt.readLine();

System.out.print("Enter Cost per month: ");

cost[i] = Integer.parseInt(inpt.readLine());

System.out.print("Enter CAvailability in percentage: ");

availability[i] = Integer.parseInt(inpt.readLine());

System.out.print("Enter Storage in MB: ");

storage[i] = Integer.parseInt(inpt.readLine());

}

// Printing the values

for (int i = 0; i < cloud\_p.length; i++) {

System.out.println("Provider :" + cloud\_p[i] + " Cost: " + cost[i] + " Availability: " + availability[i] + " Storage: " + storage[i]);

}

System.out.println("paramaters values are 1.Cost 2. Security 3.Storage");

int[] req = new int[3];

for (int i = 0; i < 3; i++)

{

System.out.println("Enter requirement :" + (i+1) );

req[i] = Integer.parseInt(inpt.readLine());

}

int[] cal1 = new int[3];

int x=0;

for (int i = 0; i < cloud\_p.length; i++){

cal1[i]= (cloud\_p.length - i) \* req[i];

cal1[i]=x + cal1[i];

x=cal1[i];

}

int temp;

String str;

for (int i = 0; i < cloud\_p.length; i++)

{

for (int j = i + 1; j < cloud\_p.length; j++)

{

if (cal1[i] < cal1[j])

{

temp = cal1[i];

cal1[i] = cal1[j];

cal1[j] = temp;

temp = cost[i];

cost[i] = cost[j];

cost[j] = temp;

temp = availability[i];

availability[i] = availability[j];

availability[j] = temp;

temp = storage[i];

storage[i] = storage[j];

storage[j] = temp;

str = cloud\_p[i];

cloud\_p[i] = cloud\_p[j];

cloud\_p[j] = str;

}

}

}

System.out.print("Ranked list for the given requirement:");

for (int i = 0; i < 3 - 1; i++)

{

System.out.print("\nProvider "+ (i+1) + " :"+ cloud\_p[i]);

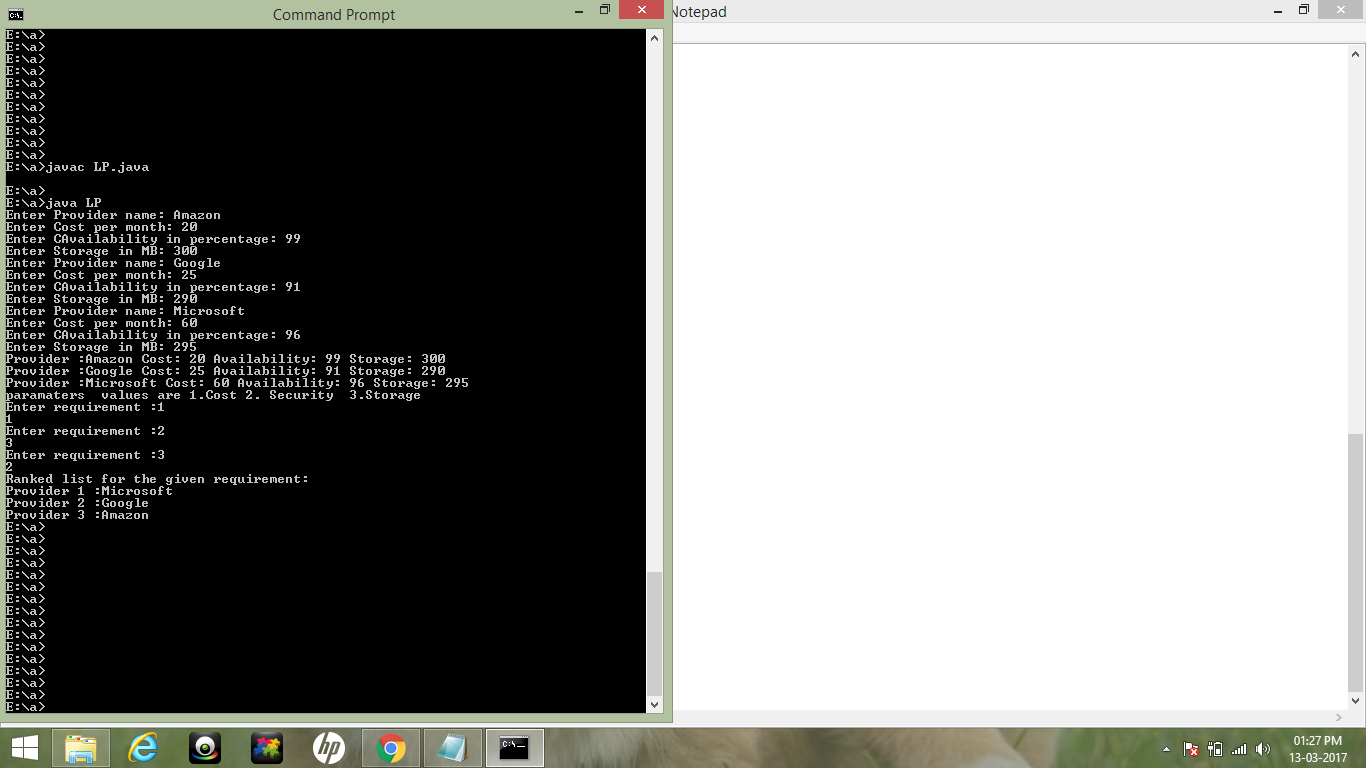
}

System.out.print("\nProvider 3 :"+ cloud\_p[3-1]);

}

}

OUTPUT



DOCUMENTATION:

**Linear programming** (LP) (also called **linear** optimization) is a method to achieve the best outcome (such as maximum profit or lowest cost) in a mathematical model whose requirements are represented by **linear** relationships.

Step 1:

Get the input from the cloud service providers in the following order:

1. name of the provider
2. cost per month
3. availability in percentage
4. Storage in MB
5. Security level

Step 2 :

The list obtained from the cloud service providers is stored in the database.

Step 3 :

1. The list of requirements from the cloud service consumers is obtained.

2. Priority for the requirements is calculated based on the consumer’s input data.

3. The Cloud service providers are ranked using the Linear programming model

a. c = p \* i

c – cloud service provider

p – priority of the consumer’s requirement

i – weight value of the particular requirement

b. Output = max(c)

max(c) – returns the maximum value of the coud service provider.

4. The service providers are ranked according to the customer’s requirement and the list of the providers is given to the customer.

GRAPH