import java.io.IOException;// to handle exceptions

import java.io.BufferedReader;//To input text from input stream

import java.io.InputStreamReader;//bridges byte stream to character stream

public class RR3

{

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

{

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

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IMPLEMENTATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

System.out.println("Enter the Time Quantum: ");

int q = Integer.parseInt(br.readLine());

/\*parseint converts input string to integer and readline converts the

integer into string\*/

System.out.println("Please enter the number of Processes: ");

int n = Integer.parseInt(br.readLine());

//process table 2d arraay

int proc[][] = new int[n + 1][4];//proc[][0] is the AT array,[][1] - RT,[][2] - WT,[][3] - TT

for(int i = 1; i <= n; i++)//filling rows==>n+1

{

System.out.println("Please enter the Burst Time for Process " + i + ": ");

proc[i][1] = Integer.parseInt(br.readLine());

}

System.out.println();

//Calculation of Total Time and Initialization of Time Chart array

int total\_time = 0;

for(int i = 1; i <= n; i++)

{

total\_time += proc[i][1];//total burst time

}

int time\_chart[] = new int[total\_time];//declare a new array which gives the Gantt chart

int sel\_proc = 1;

int current\_q = 0;

for(int i = 0; i < total\_time; i++)

{

//Assign selected process to current time in the Chart

time\_chart[i] = sel\_proc;

//Decrement Remaining Time of selected process by 1 since it has been assigned the CPU for unit of time

proc[sel\_proc][1]--;

//WT and TT Calculation

for(int j = 1; j <= n; j++)

{

if(proc[j][1] != 0)

{

proc[j][3]++;//If process has not completed execution its TT is incremented by 1

if(j != sel\_proc)//If the process has not been currently assigned the CPU its WT is incremented by 1

proc[j][2]++;

}

else if(j == sel\_proc)//This is a special case in which the process has been assigned CPU and has completed its execution

proc[j][3]++;

}

//Printing the Time Chart

if(i != 0)

{

if(sel\_proc != time\_chart[i - 1])

//If the CPU has been assigned to a different Process we need to print the current value of time and the name of

//the new Process

{

System.out.print("--" + i + "--P" + sel\_proc);

}

}

else//If the current time is 0 i.e the printing has just started we need to print the name of the First selected Process

System.out.print(i + "--P" + sel\_proc);

if(i == total\_time - 1)//All the process names have been printed now we have to print the time at which execution ends

System.out.print("--" + (i + 1));

//Updating value of sel\_proc for next iteration

current\_q++;

if(current\_q == q || proc[sel\_proc][1] == 0)//If Time slice has expired or the current process has completed execution

{

current\_q = 0;

//This will select the next valid value for sel\_proc

for(int j = 1; j <= n; j++)

{

sel\_proc++;

if(sel\_proc == (n + 1))

sel\_proc = 1;

if(proc[sel\_proc][1] != 0)

break;

}

}

}

System.out.println();

System.out.println();

//Printing the WT and TT for each Process

System.out.println("P\t WT \t TT ");

for(int i = 1; i <= n; i++)

{

System.out.printf("%d\t%3dms\t%3dms",i,proc[i][2],proc[i][3]);

System.out.println();

}

System.out.println();

//Printing the average WT & TT

float WT = 0,TT = 0;

for(int i = 1; i <= n; i++)

{

WT += proc[i][2];

TT += proc[i][3];

}

WT /= n;

TT /= n;

System.out.println("\n\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*ANSWERS\*\*\*\*\*\*\*\*\*\*\n\n");

System.out.println("The Average WT is: " + WT + "ms");

System.out.println("The Average TT is: " + TT + "ms");

}

}