UNIVERSITI MALAYA
UNIVERSITY OF MALAYA

PEPERIKSAAN IJAZAH SARJANA MUDA SAINS KOMPUTER
EXAMINATION FOR THE DEGREE OF BACHELOR OF COMPUTER SCIENCE

SESI AKADEMIK 2022/2023 : SEMESTER 1
ACADEMIC SESSION 2022/2023 : SEMESTER 1

WIX1002 : Asas-Asas Pengaturcaraan
Fundamentals of Programming

Feb 2023 Masa: 3.5 jam
Feb 2023 Time: 3.5 Hour

ARAHAN KEPADA CALON:
INSTRUCTIONS TO CANDIDATES:

Calon dikehendaki menjawab semua soalan.
Candidates should answer all questions.

(Kertas soalan ini mengandungi 4 soalan dalam 9 halaman yang dicetak)
(This question paper consists of 4 questions on 9 printed pages)

1. Tuliskan satu aturcara yang menjana nombor secara rawak dengan sempadan atas 1000. Aturcara tersebut harus memaparkan satu nombor integer yang dijana dalam susunan terbalik, dan akhirnya, mengira jumlah kesemua integer. Aturcara anda harus dilaksanakan mengikut output berikut:

   Write a program that randomly generates a number with upper bound of 1000. The program should display the integers of the generated number in a reverse order, and finally, calculate the sum of all integers. Your program should execute according to the following output:

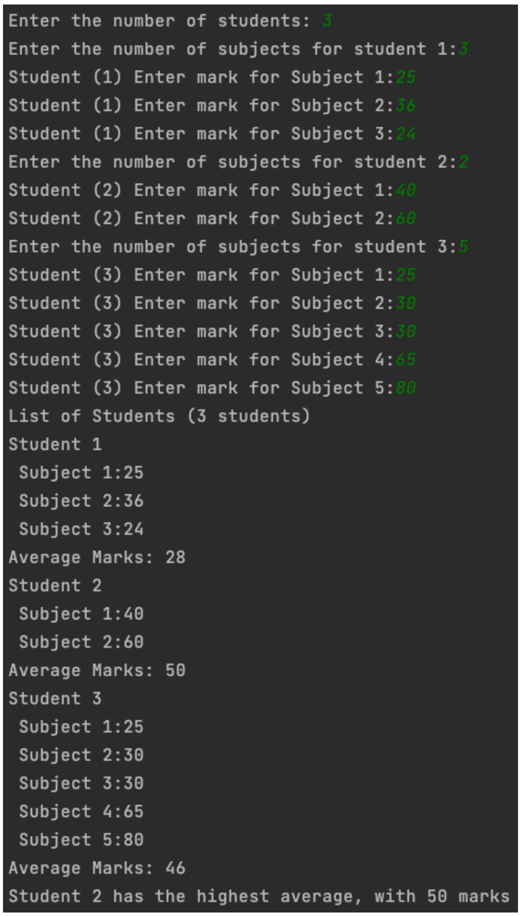
```
Generated number: 3122
Number in reverse order: 2213
The sum of all integers: 8
```

(10 markah/*marks*)

2. Tulis satu aturcara yang memenuhi hasil akhir di bawah dengan kondisi berikut:
   a) Pengguna mesti memasukkan bilangan pelajar dan subjek untuk program tersebut. Kedua dua bilangan adalah dinamik, dan pengguna boleh memasukkan nombor positf di antara [0-10].
   b) Pengguna perlu memasukkan markah bagi setiap subjek; bergantung kepada bilangan subjek yang dimasukkan untuk setiap pelajar, pengguna boleh memasukkan markah dengan nombor positif di antara [0-100].
   c) Satu mesej "Error !!!" dipaparkan apabila nombor yang dimasukkan untuk bilangan pelajar, subjek atau markah terkeluar dari julat.
   d) Hasil akhir aturcara perlu menyenaraikan semua pelajar dan markah mereka untuk setiap subjek.
   e) Aturcara juga menggira purata bagi setiap markah dan memaparkan pelajar dengan markah tertinggi pada akhir aturcara.

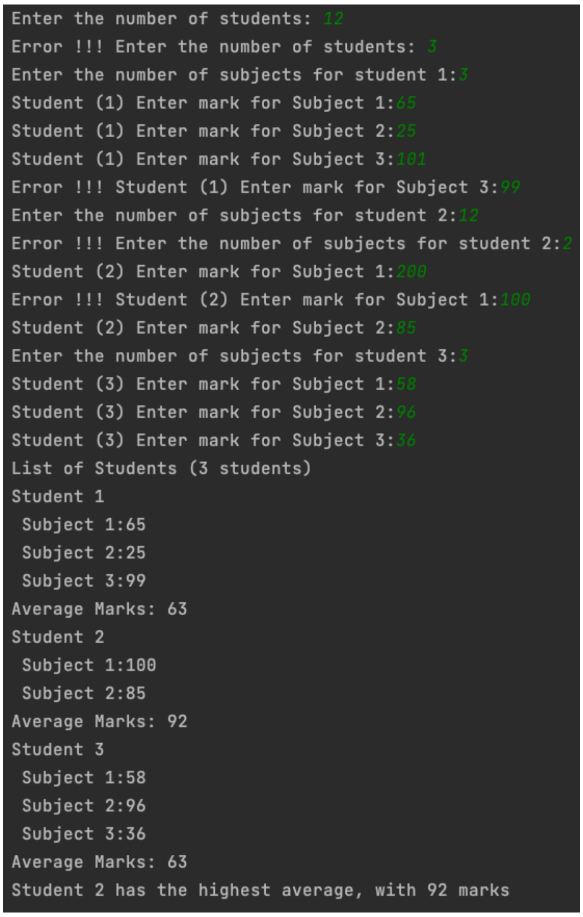
   Write a program that satisfies the output given below with the following condition:
   a) Users must enter the number of students and subjects for the program. Both numbers are dynamic, and users may enter positive numbers between [0-10].
   b) Users need to enter the marks for each subject; depending on the number of subjects entered by each student, the user may enter positive marks between [0-100].
   c) A message "Error !!!" is displayed when the number entered for the number of students, subjects, or the marks are out of bounds.
   d) The program's last output shall list all the students with their marks for each subject.
   e) The programs shall calculate the average for each mark and display the student with the highest average at the end of the program.

Contoh output 1
Sample output 1



```
Enter the number of students: 3
Enter the number of subjects for student 1:3
Student (1) Enter mark for Subject 1:25
Student (1) Enter mark for Subject 2:36
Student (1) Enter mark for Subject 3:24
Enter the number of subjects for student 2:2
Student (2) Enter mark for Subject 1:40
Student (2) Enter mark for Subject 2:60
Enter the number of subjects for student 3:5
Student (3) Enter mark for Subject 1:25
Student (3) Enter mark for Subject 2:30
Student (3) Enter mark for Subject 3:30
Student (3) Enter mark for Subject 4:65
Student (3) Enter mark for Subject 5:80
List of Students (3 students)
Student 1
 Subject 1:25
 Subject 2:36
 Subject 3:24
Average Marks: 28
Student 2
 Subject 1:40
 Subject 2:60
Average Marks: 50
Student 3
 Subject 1:25
 Subject 2:30
 Subject 3:30
 Subject 4:65
 Subject 5:80
Average Marks: 46
Student 2 has the highest average, with 50 marks
```

Contoh output 2
Sample output 2



```
Enter the number of students: 12
Error !!! Enter the number of students: 3
Enter the number of subjects for student 1:3
Student (1) Enter mark for Subject 1:65
Student (1) Enter mark for Subject 2:25
Student (1) Enter mark for Subject 3:101
Error !!! Student (1) Enter mark for Subject 3:99
Enter the number of subjects for student 2:12
Error !!! Enter the number of subjects for student 2:2
Student (2) Enter mark for Subject 1:200
Error !!! Student (2) Enter mark for Subject 1:100
Student (2) Enter mark for Subject 2:85
Enter the number of subjects for student 3:3
Student (3) Enter mark for Subject 1:58
Student (3) Enter mark for Subject 2:96
Student (3) Enter mark for Subject 3:36
List of Students (3 students)
Student 1
 Subject 1:65
 Subject 2:25
 Subject 3:99
Average Marks: 63
Student 2
 Subject 1:100
 Subject 2:85
Average Marks: 92
Student 3
 Subject 1:58
 Subject 2:96
 Subject 3:36
Average Marks: 63
Student 2 has the highest average, with 92 marks
```

(15 markah/*marks*)

3. Pelampung data ialah komponen penting yang digunakan untuk memantau dan mengumpul keadaan atmosfera dan oseanografi. Ia mengandungi penderia untuk mendapatkan data tentang angin, ombak, tekanan udara, suhu, ombak dan kemasinan, yang menghantar secara automatik ke pantai. Penderia-penderia masa nyata ini menyumbang kepada pengesanan awal, pemantauan dan ramalan kejadian cuaca, dan membolehkan kami memodelkan, memahami dan menerangkan cuaca dan iklim global.

   A data buoy is a significant component that is used to monitor and collect atmospheric and oceanographic conditions. It contains sensors to capture data on winds, waves, air pressure, temperature, swell and salinity, that transmit automatically to shore. These real-time sensors are contributes to the early detection, monitoring, and forecasting of weather occurrences, and allow us to model, understand, and describe global weather and climate.

Rajah 1: Pelampung data [sumber: *NOAA and EuroGOOS*]
Figure 1: Data buoy [source: NOAA and EuroGOOS]

Data dicatatkan dalam fail *comma-separated values* (CSV), bersama dengan lokasi, dalam latitud dan longitud, di mana pelampung diletakkan. Data disimpan dalam format di bawah:

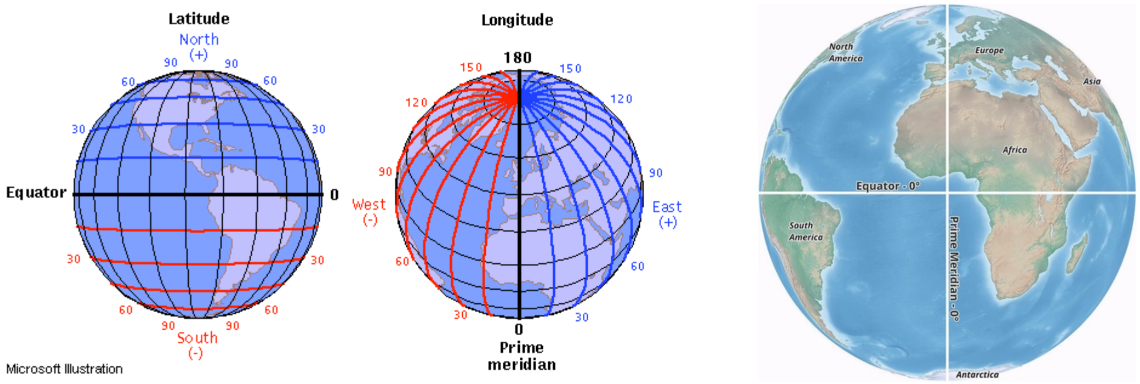
[buoy_id, latitude, longitude, air_temperature, water_temperature]

The data are logged in comma-separated values (CSV) file, together with the location, in latitude and longitude, where the buoys are deployed. The data are stored in the format below:

[buoy_id, latitude, longitude, air_temperature, water_temperature]

Lokasi pelampung direkodkan sebagai satu pasangan latitud dan longitud. Persilangan darjah 0 latitud (iaitu Khatulistiwa) dan darjah 0 longitud (iaitu Meridian Perdana) berada di tengah Lautan Atlantik. Titik ini juga dirujuk sebagai *Null Island* dan ditandai dengan pelampung yang dipanggil *Soul Buoy*.

The location of the buoy is recorded as latitude and longitude pair. The intersection of 0 degree latitude (i.e. the Equator) and 0 degree longitude (i.e. Prime Meridian) is in the middle of the Atlantic Ocean. This point also referred as the Null Island and is marked by a buoy called Soul Buoy.



Rajah 2: Latitud dan Longitud [sumber: *Illinois State University* dan *Caitlin Dempsey*]
Figure 2: Latitude and Longitude [source: Illinois State University and Caitlin Dempsey]

Sebagai contohnya, lokasi UM ialah (3.122500, 101.653038), yang bermaksud ia adalah kira-kiranya 3.122500 darjah ke utara Khatulistiwa (3° 7' 21" U), dan 101.653038 darjah ke timur Meridian Perdana (101° 39' 10.9368" E).

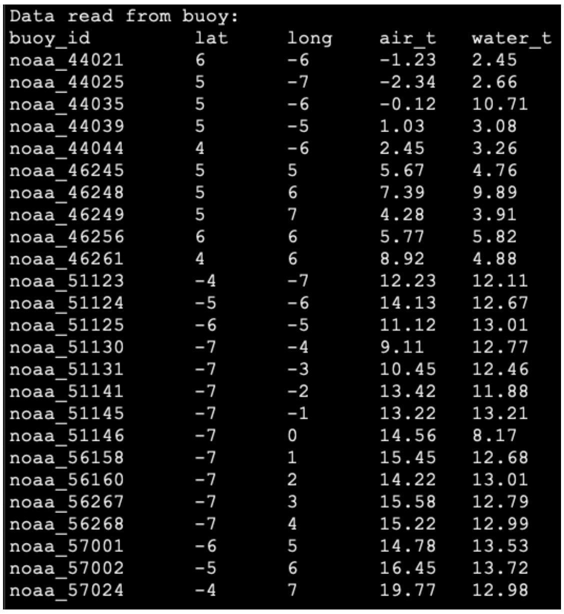
For instance, the UM location is (3.122500, 101.653038), which means it is about 3.122500 degree to the north of the Equator (3° 7' 21" N), and 101.653038 degree to the east of the Prime Meridian (101° 39' 10.9368" E).

Dalam soalan ini, anda perlu membuat satu aturcara yang bernama **OceanData.java**, yang:

a) Muatkan data daripada fail csv bernama **ocean_data.csv**. Setiap rekod disimpan sebagai objek **DataBuoy**.
b) Paparkan data yang direkodkan dalam format yang ditunjukkan dalam Rajah 3 di dalam konsol output.
c) Kira purata **suhu air** pelampung dan paparkannya dalam peta, seperti yang ditunjukkan dalam Rajah 4. **N** mewakili *Null Island*, **H** ialah lokasi dengan suhu air lebih tinggi daripada purata, **C** ialah lokasi dengan suhu air lebih rendah daripada purata, dan **A**, jika suhu yang direkodkan adalah sama dengan purata.

*In this question, you need to create a program named, **OceanData.java**, that:*

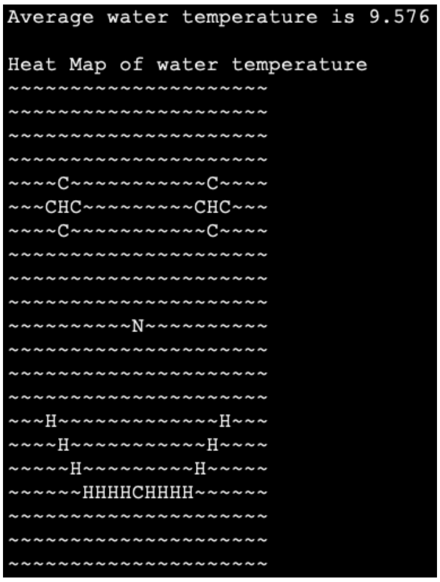
*a) Load the data from a csv file named, **ocean_data.csv**. Each of the record is stored as a **DataBuoy** object.*
b) Display the recorded data in the format shown in the Figure 3.
*c) Calculate the average **water temperature** of the buoys and display them in the map, as shown in Figure 4. **N** is representing the Null Island, **H** is the location with water temperature higher than average, **C** is the location with water temperature lower than the average, and **A**, if the recorded temperature is same as the average.*



```
Data read from buoy:
buoy_id         lat     long    air_t   water_t
noaa_44021      6       -6      -1.23   2.45
noaa_44025      5       -7      -2.34   2.66
noaa_44035      5       -6      -0.12   10.71
noaa_44039      5       -5      1.03    3.08
noaa_44044      4       -6      2.45    3.26
noaa_46245      5       5       5.67    4.76
noaa_46248      5       6       7.39    9.89
noaa_46249      5       7       4.28    3.91
noaa_46256      6       6       5.77    5.82
noaa_46261      4       6       8.92    4.88
noaa_51123      -4      -7      12.23   12.11
noaa_51124      -5      -6      14.13   12.67
noaa_51125      -6      -5      11.12   13.01
noaa_51130      -7      -4      9.11    12.77
noaa_51131      -7      -3      10.45   12.46
noaa_51141      -7      -2      13.42   11.88
noaa_51145      -7      -1      13.22   13.21
noaa_51146      -7      0       14.56   8.17
noaa_56158      -7      1       15.45   12.68
noaa_56160      -7      2       14.22   13.01
noaa_56267      -7      3       15.58   12.79
noaa_56268      -7      4       15.22   12.99
noaa_57001      -6      5       14.78   13.53
noaa_57002      -5      6       16.45   13.72
noaa_57024      -4      7       19.77   12.98
```

Rajah 3: Contoh data baca daripada fail **ocean_data.cvs**

*Figure 3: Sample data read from the **ocean_data.cvs** file*



```
Average water temperature is 9.576

Heat Map of water temperature
~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
~~~~C~~~~~~~~~~~C~~~~
~~~CHC~~~~~~~~~CHC~~~
~~~~C~~~~~~~~~~~C~~~~
~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~N~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
~~~H~~~~~~~~~~~~~H~~~
~~~~H~~~~~~~~~~~H~~~~
~~~~~H~~~~~~~~~H~~~~~
~~~~~~HHHHCHHHH~~~~~~
~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
```

Rajah 4: Contoh peta haba yang dicetak daripada data yang diperoleh daripada fail **ocean_data.cvs**

*Figure 4: Sample heat map printed from the data obtained from **ocean_data.cvs** file*

Untuk memudahkan kerja anda:

a) Latitud dan longitud ialah nombor integer, dalam julat di antara -10 hingga 10 darjah, contohnya (-1,8), (3,-5) dan (6,2).
b) Terdapat bilangan pelampung data yang tetap, iaitu 25.
c) Metod main untuk kelas **OceanData** dibekalkan di bawah.

To simply your work:

a) The latitude and longitude are integer numbers, within the range between -10 to 10 degree, e.g. (-1,8), (3,-5) and (6,2).
b) There are a fixed number of data buoys, that is 25.
*c) The main method of the **OceanData** class is given below.*

		longitude																				
		-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
latitude	10																					
	9																					
	8																					
	7																					
	6						C															
	5																	H				
	4			H																H		
	3																					
	2																					
	1																					
	0											N										
	-1																					
	-2						H															
	-3																					
	-4																					
	-5																			C		
	-6				C																	
	-7																					
	-8																C					
	-9																		C			
	-10																					

Rajah 5: Lingkungan tempat pelampung data yang digunakan di Lautan Atlantik.
Figure 5: The range where the data buoys are been deployed in the Atlantic Ocean.

Metod **Main** untuk Kelas **OceanData**
***Main** method for the **OceanData** class*

```
public class OceanData{
  public static void main (String[]args) {
    String inputFile = "ocean_data.csv";
    DataBuoy [] data = new DataBuoy[25];
    float avg_water_temp;

    loadData(inputFile, data);
    showData(data);

    avg_water_temp = calculateAvgWaterTemp(data);
    System.out.println("\nAverage water temperature is "
      + avg_water_temp);

    printMap(data, avg_water_temp);
  }
}
```

(15 markah/*marks*)

4. Salah satu tugas harian seorang pakar Pengkomputeran Berprestasi Tinggi (HPC) ialah memantau status kluster HPC. Penjadual yang digunakan dalam pusat HPC UM, SLURM, mempunyai arahan untuk memaparkan senarai maklumat nod pengkomputeran dalam kluster HPC.

   One of the daily jobs of a High Performance Computing (HPC) specialist is to monitor the status of the HPC cluster. The scheduler used in UM HPC centre, SLURM, has a command to display a list of computing nodes information in the HPC cluster.

```
NodeName=cpu01 Arch=x86_64 CpuBind=ldoms CoresPerSocket=8
   CPUAlloc=16 CPUTot=64 CPULoad=64.02
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=(null)
   NodeAddr=cpu01 NodeHostName=cpu01 Version=21.08.8-2
   OS=Linux 3.10.0-1160.25.1.el7.x86_64 #1 SMP Wed Apr 28 21:49:45 UTC 2021
   RealMemory=240000 AllocMem=35840 FreeMem=230493 Sockets=8 Boards=1
   State=ALLOCATED ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=cpu-opteron
   BootTime=2022-04-27T11:50:08 SlurmdStartTime=2022-05-11T07:47:10
   LastBusyTime=2022-12-05T23:38:26
   CfgTRES=cpu=64,mem=240000M,billing=16000
   AllocTRES=cpu=16,mem=35G
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s

NodeName=cpu04 Arch=x86_64 CpuBind=ldoms CoresPerSocket=8
   CPUAlloc=64 CPUTot=64 CPULoad=63.57
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=(null)
   NodeAddr=cpu04 NodeHostName=cpu04 Version=21.08.8-2
   OS=Linux 3.10.0-1160.25.1.el7.x86_64 #1 SMP Wed Apr 28 21:49:45 UTC 2021
   RealMemory=240000 AllocMem=239616 FreeMem=201636 Sockets=8 Boards=1
   State=ALLOCATED ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=cpu-opteron
   BootTime=2022-04-15T10:10:38 SlurmdStartTime=2022-05-11T07:47:09
   LastBusyTime=2022-12-04T22:21:28
   CfgTRES=cpu=64,mem=240000M,billing=16000
   AllocTRES=cpu=64,mem=234G
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s

[. . . continue]
```

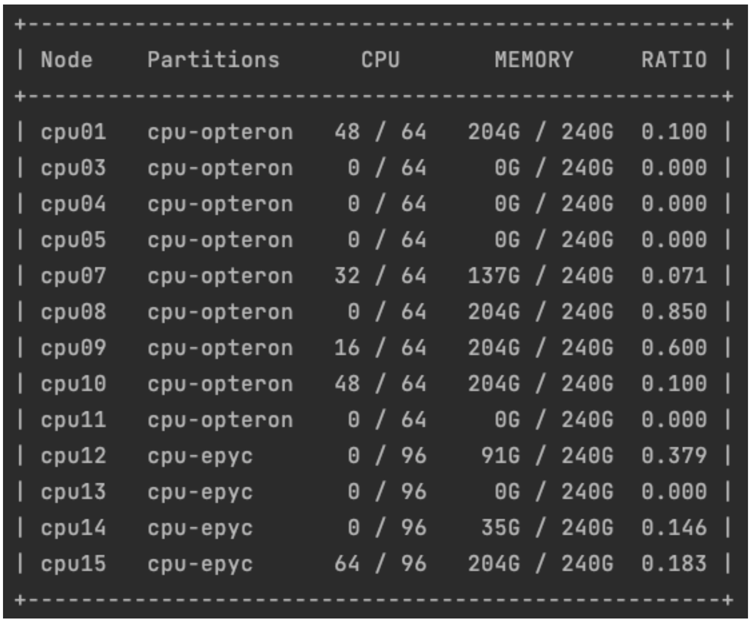
   Pemetaan untuk maklumat nod yang diperlukan dalam soalan ini adalah:
   - *NodeName*: nama nod
   - *CPUTot*: jumlah bilangan CPU yang sedia ada
   - *CPUAlloc*: bilangan CPU yang telah diperuntukkan
   - *RealMemory*: jumlah bilangan RAM dalam GB
   - *AllocMem*: bilangan RAM yang telah diperuntukkan
   - *Partition*: *partition* di mana nod telah ditugaskan

   Mapping for nodes information which is needed in this question are:
   - *NodeName: name of the node*
   - *CPUTot: total number of available CPU*
   - *CPUAlloc: number of CPUs that have been allocated*
   - *RealMemory: total number of RAM in GB*
   - *AllocMem: number of RAM that have been allocated*
   - *Partitions: the partitions where the node has been assigned*

Diberikan contoh fail input, **nodelist.txt**, tulis aturcara untuk mengekstrakkan maklumat dan memaparkannya dalam bentuk sebuah jadual, seperti yang ditunjukkan dalam contoh output dalam Rajah 6. Ia harus mengandungi nama nod, partition, bilangan CPU, MEMORY dalam GB dan nisbah kecekapan peruntukan. Nisbah kecekapan ialah nombor bukan negatif yang dikira menggunakan formula di bawah:

Given a sample input file, ***nodelist.txt****, write a program to extract the information and display them in the form of a table, as shown in the sample output in Figure 6. It should contain the name of the node, the partitions, the number of CPUs, the MEMORY in GB, and the allocation efficiency ratio. The efficiency ratio is a non-negative number calculated using the formula below:*

$$Efficiency\, ratio = \left| \frac{Free\ CPU}{Total\ CPU} - \frac{Free\ RAM}{Total\ RAM} \right|$$



```
+--------------------------------------------------+
| Node    Partitions      CPU       MEMORY     RATIO |
+--------------------------------------------------+
| cpu01   cpu-opteron   48 / 64   204G / 240G  0.100 |
| cpu03   cpu-opteron    0 / 64     0G / 240G  0.000 |
| cpu04   cpu-opteron    0 / 64     0G / 240G  0.000 |
| cpu05   cpu-opteron    0 / 64     0G / 240G  0.000 |
| cpu07   cpu-opteron   32 / 64   137G / 240G  0.071 |
| cpu08   cpu-opteron    0 / 64   204G / 240G  0.850 |
| cpu09   cpu-opteron   16 / 64   204G / 240G  0.600 |
| cpu10   cpu-opteron   48 / 64   204G / 240G  0.100 |
| cpu11   cpu-opteron    0 / 64     0G / 240G  0.000 |
| cpu12   cpu-epyc       0 / 96    91G / 240G  0.379 |
| cpu13   cpu-epyc       0 / 96     0G / 240G  0.000 |
| cpu14   cpu-epyc       0 / 96    35G / 240G  0.146 |
| cpu15   cpu-epyc      64 / 96   204G / 240G  0.183 |
+--------------------------------------------------+
```

Rajah 6: Contoh output dalam bentuk sebuah jadual
Figure 6: Sample output in the form of a table

(10 markah/*marks*)

TAMAT / END