

UNIVERSITI MALAYA
UNIVERSITY OF MALAYA

PEPERIKSAAN IJAZAH SARJANA MUDA TEKNOLOGI MAKLUMAT
EXAMINATION FOR THE DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY

SESI AKADEMIK 2018/2019 : SEMESTER II
ACADEMIC SESSION 2018/2019 : SEMESTER II

WIB1002 : Struktur Data
Data Structure

Jun 2019
June 2019

Masa: 3 jam 30 minit
Time: 3 hours 30 minutes

ARAHAN KEPADA CALON:
INSTRUCTIONS TO CANDIDATES:

Calon dikehendaki menjawab **SEMUA** soalan (50 markah).
Answer **ALL** questions (50 marks).

(Kertas soalan ini mengandungi 4 soalan dalam 10 halaman yang dicetak)
(This question paper consists of 4 questions on 10 printed pages)

1. Bina satu projek "Netbeans" bernama "Q1_NomborMatrikAnda" untuk menyimpan semua fail kod anda untuk Soalan 1. Taipkan nombor matrik anda sebagai baris pertama bagi setiap fail.

Create a Netbeans project entitled "Q1_YourMetricNumber" to store all your code files for Question 1. Type your metric number as the first line of each file.

Dalam soalan ini, anda akan menulis satu aturcara yang mencipta dua senarai berpaut, dan kemudian menggabungkan kedua-dua senarai ini menjadi satu senarai berpaut baru.

In this question, you are going to write a program that creates two linked-lists and then combine these two lists into one new linked-list.

- a) Tulis satu kelas Java *Node* yang menyimpan dua atribut dan satu pembina. Atribut pertama ialah satu atribut berjenis generik dan atribut kedua ialah satu petunjuk ke *Node* yang lain. Pembina ini menerima satu parameter berjenis generik.

Write a Java class Node with two attributes and a constructor. The first attribute is an attribute of generic type and the second attribute is a pointer to another Node. The constructor receives one parameter of generic type.

(1 markah/mark)

- b) Fail kelas Java bernama *MyLinkedList* mempunyai *Node* yang dibangunkan dalam bahagian (a) sebagai atribut. Bangunkan fail kelas ini dengan atribut-atribut, pembina dan metod-metod berikut:
 - i) Isytiharkan atribut-atribut *head*, *tail* dan *size*.
 - ii) Satu pembina lalai yang menetapkan *head*, *tail* dan *size* kepada nilai-nilai yang sesuai.
 - iii) Satu metod *getSize* yang memulangkan saiz senarai berpaut berkenaan.
 - iv) Satu metod *addFirst* yang menambah satu nod ke posisi pertama senarai berpaut berkenaan dan tidak memulangkan apa-apa.
 - v) Satu metod *addLast* yang menambah satu nod ke posisi terakhir senarai berpaut berkenaan dan tidak memulangkan apa-apa.
 - vi) Satu metod *removeFirst* yang membuang satu nod di posisi pertama senarai berpaut berkenaan dan memulangkan elemen yang disimpan dalam nod tersebut.
 - vii) Menimpa metod *toString* supaya saiz senarai berpaut berkenaan, berikutan dengan elemen dalam setiap nod akan dicetak mengikut susunan. Contohnya, jika senarai berpaut mempunyai 3 nod yang menyimpan *string* "Hello", "World" and "!" masing-masing, pemanggilan metod *toString* mencetak:

```
[size=3] >> Hello >> World >> !
```

Java class file *MyLinkedList* has *Node* developed in part (a) as attribute. Develop this class file with the following attributes, constructor and methods:

- i) Declare 3 attributes named *head*, *tail* and *size*.
- ii) A default constructor that set the *head*, *tail* and *size* to appropriate values.
- iii) A *getSize* method that returns the size of the linked-list.
- iv) An *addFirst* method that add a node to the first position of the linked-list and return nothing.
- v) An *addLast* method that add a node to the last position of the linked-list and return nothing.
- vi) A *removeFirst* method that remove the first node in the linked-list, and return the element stored in the node.
- vii) Override the *toString* method so that the size of the list, follows by element in each node will be printed in sequence. For example, if the linked-list has 3 nodes storing strings "Hello", "World" and "!" respectively, a call to this *toString* method prints:

```
[size=3] >> Hello >> World >> !
```

(7 markah/marks)

- c) Dalam file ketiga, bangunkan satu aturcara yang mengisytiharkan tiga *MyLinkedList* untuk menyimpan Integer, iaitu *list1*, *list2* dan *newList*. *list1* ialah satu senarai dengan 5 nombor dalam susunan meningkat: "2", "10", "38", "41" dan "51"; manakala *list2* mempunyai 7 nombor dalam susunan meningkat yang sama: "5", "12", "18", "21", "35", "41" dan "56". Cetak *list1*, *list2* dan *newList* yang akan dijana dalam bahagian (d) soalan ini. Contoh output adalah seperti Rajah 1 di bawah:

In the third file, develop a program that declares three MyLinkedList to store Integers, namely list1, list2 and newList. The list1 is a list with 5 numbers in incremental order: "2", "10", "38", "41" and "51"; whereas list2 with 7 numbers in the same incremental order: "5", "12", "18", "21", "35", "41" and "56". Print out list1, list2 and newList which will be generated in part (d) of this question.

Example of the output is as Figure 1 below:

```
[size=5] >> 2 >> 10 >> 38 >> 41 >> 51
[size=7] >> 5 >> 12 >> 18 >> 21 >> 35 >> 41 >> 56
The newList is:
[size=12] >> 2 >> 5 >> 10 >> 12 >> 18 >> 21 >> 35 >> 38 >> 41 >>
41 >> 51 >> 56
```

Rajah 1 / Figure 1

(2 markah/marks)

- d) Dalam fail yang anda bangunkan dalam bahagian (c), cipta satu metod *combine* yang menggunakan metod-metod dalam kelas *MyLinkedList* untuk mengintegrasikan dua senarai. Kalau kedua-dua senarai input mempunyai integer disusun dalam urutan menaik, senarai baru yang dijana adalah satu senarai dengan semua integer dalam kedua-dua senarai dalam urutan menaik juga.

In the file you developed in part (c), create a method `combine` that make use of methods in `MyLinkedList` class to integrate two lists. If both the input lists have integers sorted in ascending order, the new list generated is a list with all the integers in both lists in ascending order as well.

(3 markah/marks)

Markah tidak akan diberi bagi apa-apa pelaksanaan yang menggunakan kelas "`LinkedList`" daripada "`Java API Library`".

No mark will be given for any implementation using the existing `LinkedList` class from the Java API Library.

2. Bina satu projek "Netbeans" bernama "Q2_NomborMatrikAnda" untuk menyimpan semua fail kod anda untuk Soalan 2. Taipkan nombor matrik anda sebagai baris pertama bagi setiap fail.

Create a Netbeans project entitled "Q2_YourMetricNumber" to store all your code files for Question 2. Type your metric number as the first line of each file.

Dalam program ini, anda dikehendaki untuk menyusun-semula senarai data jualan, yang mengandungi maklumat mengenai *Region*, *Sales Representative Name* dan *Sales Amount*. Anda perlu membina dua kelas, termasuk *SalesList* dan *SortSalesList*.

Dalam *SalesList*, anda perlu mengisytiharkan tiga pembolehubah, mengekodkan pembina lalai dan pembina yang mentakrifkan tiga pembolehubah, serta metod *toString()* yang memaparkan output dalam bentuk yang ditunjukkan.

(2 markah)

Dalam *SortSalesList*, anda dikehendaki untuk menggunakan *Comparable* dalam menyiapkan tugas-tugas berikut:

- a) Menyusun-semula data *Region* dengan menggunakan *Selection Sort* dalam susunan menurun.
(3 markah)
- b) Menyusun-semula data *Sales Amount* dengan menggunakan *Insertion Sort* dalam susunan menaik.
(2.5 markah)
- c) Menyusun-semula data *Sales Representative Name* dengan menggunakan *Bubble Sort* dalam susunan menaik.
(2.5 markah)

Tuliskan *main* metod yang memanggil metod-metod yang berkenaan untuk memaparkan hasil susun-semula tersebut.

(2 markah)

Data asal dan output sampel adalah ditunjukkan dalam Rajah 2 di bawah:

You are asked to sort a list of sales data, which consisting of Region, Sales Representative Name and Sales Amount. You need to create two classes, SalesList and SortSalesList.

In SalesList, you need to declare the three relevant variables, code the default no-arg constructor and constructor which defines three variables, and the toString() method that structure the display of the output.

(2 marks)

In SortSalesList, you are required to use Comparable to:

a) Sort Region data using Selection Sort in descending order.

(3 marks)

b) Sort Sales Amount using Insertion Sort in ascending order.

(2.5 marks)

c) Sort Sales Representative Name using Bubble Sort in ascending order.

(2.5 marks)

Then, write a main method which calls the relevant methods to display the sorting results.

(2 marks)

The original data and sample output are given in Figure 2 below:

Original Data:

Asia	450	Ahmad
Europe	518	Hemish
Europe	650	Banny
Greenland	200	Ain
Asia	390	Ali
Greenland	670	Ben
Asia	790	Lee

Sorting Result on Region using SelectionSort:

Greenland	200	Ain
Greenland	670	Ben
Europe	650	Banny
Europe	518	Hemish
Asia	390	Ali
Asia	450	Ahmad
Asia	790	Lee

Sorting Result on Sales using InsertionSort:

Greenland	200	Ain
Asia	390	Ali
Asia	450	Ahmad
Europe	518	Hemish
Europe	650	Banny
Greenland	670	Ben
Asia	790	Lee

Sorting Result on SalesRep using BubbleSort:

Asia	450	Ahmad
Greenland	200	Ain
Asia	390	Ali
Europe	650	Banny
Greenland	670	Ben
Europe	518	Hemish
Asia	790	Lee

Rajah 2 / Figure 2

3. Bina satu projek "Netbeans" bernama "Q3_NomborMatrikAnda" untuk menyimpan semua fail kod anda untuk Soalan 3. Taipkan nombor matrik anda sebagai baris pertama bagi setiap fail.

Create a Netbeans project entitled "Q3_YourMetricNumber" to store all your code files for Question 3. Type your metric number as the first line of each file.

Kejohanan *Triwizard* adalah pertandingan ajaib yang diadakan antara tiga sekolah sihir terbesar di Eropah. *Harry Potter* telah meluluskan dua tugas pertama. Tugas akhir yang paling berbahaya, *maze Triwizard* adalah halangan akhir yang mesti dihadapi. Profesor *Dumbledore* telah meminta anda untuk membantu *Harry*, bukan dengan menggunakan tongkat sihir anda, tetapi kemahiran komputer anda.

Anda perlu membangunkan program untuk mengesan pergerakan *Harry* dalam *maze*. Program ini pertama kali membaca Peta *Marauders*, yang merupakan fail teks bernama "maze.txt", dan memaparkannya.

```
1 0 1 1 0
1 1 1 0 1
0 1 0 1 C
1 1 0 1 0
0 1 1 1 0
```

1 menandakan laluan, 0 bermakna jalan mati, dan C ialah Piala *Triwizard*.

Harry akan menggunakan algoritma mudah untuk mencari Piala *Triwizard* yang tersembunyi dalam *maze*, cuba dan ralat. Dia akan memasuki *maze* di lokasi (0,0) dan mula meneroka semua jalan yang mungkin ada. Sekiranya *Harry* bertemu dengan jalan mati, dia akan kembali ke blok *maze* sebelumnya.

Sekiranya *Harry* berjaya menemui piala, program anda harus menunjukkan lokasi piala, serta jalan kepadanya, seperti di bawah:

The Triwizard Maze:

```
1 0 1 1 0
1 1 1 0 1
0 1 0 1 C
1 1 0 1 0
0 1 1 1 0
```

Hurray!!! Harry found the Triwizard Cup at (2,4)

This is the path Harry has taken: (0,0) > (1,0) > (1,1) > (2,1) > (3,1) > (4,1) > (4,2) > (4,3) > (3,3) > (2,3) > (2,4)

Jika tidak, program itu harus memberi amaran kepada kami:

The Triwizard Maze:

```
1 0 1 1 0
1 1 1 0 1
0 1 0 C C
1 1 0 1 0
0 1 1 1 1
```

Oh no!!! Harry couldn't find the Triwizard Cup!
Someone quickly gets Professor Dumbledore.

Tugas anda:

- a) Ini adalah masalah *backtracking* yang biasa, di mana anda boleh menggunakan timbunan untuk menyimpan semua koordinat (baris, lajur) dan arah yang mungkin (iaitu atas, kiri, bawah dan kanan), sebagai nod, dan menolaknya ke timbunan. Melaksanakan kelas tindakan yang generik. Tiada markah akan diberikan jika anda menggunakan *ADT Stack* dalam *Java Collection Framework*.

(5 markah)

- b) Tulis program seperti yang diterangkan di atas dengan menggunakan timbunan yang telah anda laksanakan dalam di atas. Anda tidak perlu bimbang sama ada ia adalah jalan terpendek, sekiranya terdapat lebih daripada satu.

(10 markah)

The Triwizard Tournament is a magical contest held between the three largest wizarding schools of Europe. Harry Potter has passed the first two tasks. The most dangerous final task, the Triwizard Maze is the final obstacle he must face. Professor Dumbledore has demanded you to help the poor Harry, not using your magic wand, but your computer skills.

You will need to develop a program to track Harry's movement in the maze. The program first read the Marauders Map, which is a text file named "maze.txt", and display it.

```
1 0 1 1 0
1 1 1 0 1
0 1 0 1 C
1 1 0 1 0
0 1 1 1 0
```

1 denotes a path, 0 means it is a dead end, and C is the of the Triwizard Cup.

Harry will use a simple algorithm to find the Triwizard Cup hidden in the maze, try and error. He'll enter the maze at location (0,0) and start to explore all possible. If Harry meets a dead end, he will retract back to the previous block of the maze.

If Harry managed to find the cup, your program should show the location of the cup, as well as the path to it, as below:

The Triwizard Maze:

```
1 0 1 1 0
1 1 1 0 1
0 1 0 1 C
1 1 0 1 0
0 1 1 1 0
```

Hurray!!! Harry found the Triwizard Cup at (2,4)

This is the path Harry has taken: (0,0) > (1,0) > (1,1) > (2,1) > (3,1) > (4,1) > (4,2) > (4,3) > (3,3) > (2,3) > (2,4)

Otherwise, the program should warn us:

```
The Triwizard Maze:
```

```
1 0 1 1 0
1 1 1 0 1
0 1 0 0 0
1 1 0 1 0
0 1 1 1 1
```

```
Oh no!!! Harry couldn't find the Triwizard Cup!
Someone quickly gets Professor Dumbledore.
```

Your task:

- a) *This is a typical backtracking problem, where you can use a stack to keep all of the possible coordinates (row, column) and directions (i.e. up, left, down and right), as a node, and push it onto the stack. Implement a generic stack class. No marks will be awarded if you use the ADT Stack in the Java Collection Framework.*

(5 marks)

- b) *Write the program as described above using the stack that you have implemented in above. You do not need to worry about whether it is a shortest path, if there exist more than one.*

(10 marks)

4. a) Bina satu projek "Netbeans" bernama "Q4a_NomborMatrikAnda" untuk menyimpan semua fail kod anda untuk Soalan 4(a). Taipkan nombor matrik anda sebagai baris pertama bagi setiap fail.

Create a Netbeans project entitled "Q4a_YourMetricNumber" to store all your code files for Question 4(a). Type your metric number as the first line of each file.

Tulis satu program Java untuk mencetak "Hello-World" tanpa menggunakan semicolon.

(4 markah)

Write a Java program to print "Hello-World" without using semicolon.

(4 marks)

- b) Taip jawapan anda untuk Soalan 4(b) di dalam satu editor teks dan namakannya "Q4b_NomborMatrikAnda.txt".

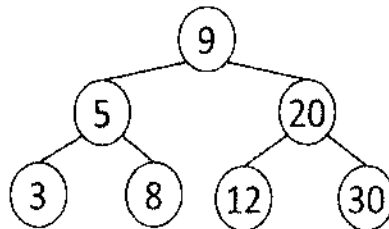
Type your answer for Question 4(b) in a text editor and name it "Q4b_NomborMatrikAnda.txt".

Bagikan "preorder", "inorder" dan "postorder" traversal untuk pokok yang berikut.

(6 marks)

Give the preorder, inorder and postorder traversal of the following tree.

(6 marks)



TAMAT
END