

UNIVERSITI MALAYA
UNIVERSITY OF MALAYA

PEPERIKSAAN IJAZAH SARJANA MUDA SAINS KOMPUTER / SARJANA MUDA
TEKNOLOGI MAKLUMAT
*EXAMINATION FOR THE DEGREE OF BACHELOR OF COMPUTER SCIENCE / BACHELOR
OF INFORMATION TECHNOLOGY*

SESI AKADEMIK 2020/2021 : SEMESTER II
ACADEMIC SESSION 2021/2021 : SEMESTER II

WIA1002/WIB1002 : Struktur Data
Data Structure

Jun 2021
June 2021

Masa: 3 jam
Time: 3 hours

ARAHAN KEPADA CALON :
INSTRUCTIONS TO CANDIDATES :

Jawab SEMUA soalan (50 markah).

Answer ALL questions (50 marks).

(Kertas soalan ini mengandungi 8 soalan dalam 15 halaman yang dicetak)
(This question paper consists of 8 questions on 15 printed pages)

Soalan / Question 1

Taip jawapan untuk soalan ini dalam borang yang diberi.
Type the answers to this question in the form provided.

Berikut adalah satu program yang tidak lengkap untuk mencetak nombor yang lebih besar di antara x dan y dengan menggunakan *Generics*. Dengan menjawab soalan (a), (b) dan (c) seperti yang ditunjukkan dalam bahagian komen, jadikan kod ini dapat dilarikan tanpa kesilapan. Jawab “*nothing*” jika tiada apa-apa diperlukan untuk satu sub soalan.

The following is an incomplete program to print out the larger number between x and y using Generics. By answering questions (a), (b) and (c) as indicated in the comment section, make the code runs without error. Answer “nothing” if nothing is required for a subquestion.

(3 markah / marks)

```

1 public class UseGenerics_____ {           // question (a)
2     private T a;
3
4     public _____ {                       // question (b)
5         if (p.compareTo(q)>0)
6             a = p;
7         else
8             a = q;
9     }
10
11    public T getA() {
12        return a;
13    }
14
15    public static void main(String args[]) {
16        int x = 3;
17        int y = 5;
18
19        _____                          // question (c)
20        System.out.println("Max is "
21        + run1.getA());
22    }
23 }

```

- d) Jika kod di bawah ditambah ke baris 21, nyatakan apakah impak yang akan terjadi? Samada satu *compilation error*, satu *runtime error*, cetak “World”, atau cetak sesuatu yang lain?

If the following code is inserted to line 21, specify what the impact will be, whether a compilation error, a runtime error, print out “World”, or print out something else?

```
System.out.println(new UseGenerics("Hello", "World"));
```

(1 markah / mark)

Soalan / Question 2

Ketiga-tiga fail `.java` dalam soalan ini terlampir. Untuk menjawab soalan ini, lengkapkan `SpecialLL.java` and `PrintPages.java` seperti yang diarahkan dalam soalan, dan serahkan kedua-dua fail `.java` sahaja.

All the three .java files in this question are attached. To answer this question, complete the SpecialLL.java and PrintPages.java as instructed in the question, and submit both .java files only.

Berikut adalah kelas `Node` yang dibina untuk suatu senarai pautan.

The following is a Node class built for a linked-list.

```

1 public class Node<E> {
2     E element;
3     Node<E> next;
4     public Node(E item) {
5         element = item;
6         next = null;
7     }
8 }
```

Kelas `SpecialLL` adalah satu senarai pautan tidak lengkap yang menggunakan kelas `Node` di atas.

The class SpecialLL is an incomplete linked-list that utilizes the Node class above.

```

1 public class SpecialLL<E> {
2     Node<E> head, tail;
3     int size = 0;
4
5     public SpecialLL() {}
6
7     public int getSize() {
8         return size;
9     }
10
11    public void addFirst(E e) {
12        Node<E> newNode = new Node<E>(e);
13        newNode.next = head;
14        head = newNode;
15        size++;
16        if (tail == null)
17            tail = head;
18    }
19 }
```

```

20     public void addLast(E e) {
21         Node<E> newNode = new Node<E>(e);
22         if (tail == null) {
23             head = tail = newNode;
24         }
25         else {
26             tail.next = newNode;
27             tail = tail.next;
28         }
29         size++;
30     }
31
32     public void print() {
33         Node<E> current = head;
34         for(int i=0; i<size; i++) {
35             System.out.print(current.element + " ");
36             current=current.next;
37         }
38         System.out.println();
39     }
40
41 }

```

Dengan tidak mengubahsuai kelas *Node*, lengkapkan kelas *SpecialLL* dengan:

Without changing the Node class, complete the SpecialLL class by:

- Tambahkan satu kaedah *printOdd* yang tidak memulangkan apa-apa dan tidak menerima sebarang hujahan. Kaedah ini membaca semua kandungan nod bernombor ganjil, dan cetak setiap satu kandungan diikuti dengan satu petak kosong. Anggapkan nod pertama (yang berindex 0) dinomborkan 1. Rujuk Gambarajah 1 untuk penerangan. Dalam kes yang digambarkan dalam Gambarajah 1, kaedah ini harus mencetak "a c e" sahaja.

Append a method `printOdd` that returns nothing and accepts no argument. The method reads all the content of odd numbered nodes, and prints out each content followed by a blank space. Assume the first node (which has index 0) is numbered as 1. Refer to Figure 1 for clarification. In the case illustrated by Figure 1, the method should print out "a c e" only.

(2 markah / marks)



Gambarajah / Figure 1

- b) Tambahkan satu kaedah *getEven* yang memulangkan satu *ArrayList* dan tidak menerima sebarang hujahan. *ArrayList* yang dipulangkan harus menyimpan semua kandungan nod yang bernombor genap dalam susunan terbalik. Anggapkan nod pertama (yang berindex 0) dinomborkan 1. Rujuk Gambarajah 1 untuk penerangan. Dalam kes yang digambarkan dalam Gambarajah 1, *ArrayList* ini harus menyimpan "f", "d" dan diikuti dengan "b" sahaja.

Append a method `getEven` that returns an `ArrayList` and accepts no argument. The `ArrayList` returned should store all the content of even numbered nodes in reverse order. Assume the first node (which has index 0) is numbered as 1. Refer to Figure 1 for clarification. In the case illustrated by Figure 1, the `ArrayList` should store "f", "d" and followed by "b" only.

(3 markah / marks)

Terdapat satu kelas *PrintPages* seperti di bawah. Dalam baris 10, program ini membaca satu nombor bulat sebagai mukasurat permulaan sebuah buku, dan dalam baris 12, ia membaca satu lagi nombor bulat sebagai nombor mukasurat yang hendak dicetak.

There is a class `PrintPages` as follows. In line 10, this program reads an integer as starting page of a book, and in line 12, it reads another integer as number of pages to print.

```

1  import java.util.Scanner;
2  import java.util.ArrayList;
3
4  public class PrintPages {
5      public static void main(String[] args) {
6          SpecialLL<Integer> bookPage = new SpecialLL<>();
7          ArrayList<Integer> selectedList = new ArrayList<>();
8          Scanner input = new Scanner(System.in);
9          System.out.print("Enter the starting page: ");
10         int start = input.nextInt();
11         System.out.print("Enter number of pages: ");
12         int page = input.nextInt();
13         // add 2 lines of code here to
14         //     insert values to bookPage
15         selectedList=bookPage.getEven();
16
17         // insert multiple lines of code here
18
19     }

```

Lengkapkan program ini dengan:

Complete this program by:

- c) Tambah dua baris kod di baris 13 dan 14 supaya program tambah semua nombor mukasurat ke *bookPage* mengikut susunan. Contohnya, jika input *start* dan

page adalah 21 dan 6 masing-masing, kedua-dua baris kod ini harus menambah 21, 22, 23, 24, 25 dan 26 ke `bookPage` dalam susunan tersebut.

Add 2 lines of code at line 13 and 14 so that the program adds all the page numbers to `bookPage` in sequence. For example, if the inputs of `start` and `page` are 21 and 6 respectively, both of these 2 lines of code should add 21, 22, 23, 24, 25 and 26 to `bookPage` in the sequence.

(1 markah / mark)

- d) Tambah beberapa baris kod bermula dari baris 17 supaya program ini mencetak nombor mukasurat ganjil dan genap dalam dua baris yang berbeza, dan setiap nombor harus dipisahkan dengan satu petak kosong. Baris kedua harus bermula dengan satu petak kosong. Rujuk output-output sampel yang ditunjukkan dalam Gambarajah 2.

Insert multiple lines of code from line 17 onwards so that the program prints odd and even page numbers in two different lines, and each number is separated by a space. The second line should start with a leading space. Refer to the sample outputs shown in Figure 2.

(2 markah / marks)

```
Enter the starting page: 1
Enter number of pages: 6
1 3 5
 2 4 6

Enter the starting page: 2
Enter number of pages: 6
2 4 6
 3 5 7
```

Gambarajah / Figure 2

Soalan / Question 3

Taip jawapan untuk soalan ini dalam borang yang diberi.

Type the answers to this question in the form provided.

Berikut adalah satu program yang menggunakan *Stack*.

The following is a program using Stack.

```
1 import java.util.Stack;
2
3 public class StackApp {
4
5     public static void main(String args[])    {
6         Stack<Integer> st = new Stack<>();
7         Integer[] x = { 4, 5, 6, 2, 9, 5, 1, 4, 1, 3};
8         for (int i=0; i<x.length; i++)
9             st.push(x[i]);
10        for (int j=0; j<3; j++)                // Question (a)
11            System.out.print(st.pop());
12        System.out.println();
13        System.out.println(st.peek());        // Question (b)
14        for (int k=0; k<3; k++)
15            st.pop();
16        System.out.println(st.pop());        // Question (c)
17    }
18 }
```

- a) Nyatakan output baris 10 dan 11.

Specify the output of lines 10 and 11.

- b) Nyatakan output baris 13.

Specify the output of line 13.

- c) Nyatakan output baris 16.

Specify the output of line 16.

(3 markah / marks)

Soalan / Question 4

Untuk menjawab soalan ini, serahkan fail *.java* sahaja.

To answer this question, submit the .java files only.

Anda telah ditugaskan untuk membina satu sistem penurunan bagi pengangkutan bas automatik di dalam sesebuah bandar pintar. Pada setiap hari bekerja, pukul 7 pagi, bas automatik akan mengambil sekumpulan penumpang dari tempat pengambilan di dalam bandar pintar tersebut. Sewaktu kemasukan, bas automatik akan menerima maklumat penumpang yang akan digunakan dalam menyusun urutan dalam sistem penurunan tersebut.

Maklumat penumpang yang diterima menunjukkan nama penumpang dan lokasi tempat kerjanya, sebagai contoh, {"Ally", "Road 3.5"} menyatakan bahawa nama penumpang ini adalah Ally dan lokasi tempat kerjanya terletak dalam jarak 3.5 KM daripada pintu masuk bandar pintar tersebut.

Tempat pengambilan dalam bandar pintar adalah di lokasi *Road 0.2* (bermaksud berjarak 0.2 KM daripada pintu masuk bandar pintar). Bas automatik akan berhenti pada setiap sela berjarak 1.0 KM, bermula daripada tempat pengambilan. Bandar pintar ini mempunyai 5 perhentian bas (tidak termasuk tempat pengambilan), bernama Perhentian A, B, C, D, dan E. Bas automatik akan menurunkan setiap penumpang di bas perhentian yang terdekat dengan tempat kerjanya.

Dengan menggunakan *PriorityQueue* dan *custom Comparator*, bina sebuah sistem penurunan untuk mengira dan memaparkan penumpang-penumpang yang berkenaan pada setiap bas perhentian, sekiranya maklumat penumpang yang berikut diberikan:

"Susan", "Road 0.9"
 "Jack", "Road 5.5"
 "Benjamin", "Road 1.6"
 "Rachel", "Road 3.8"
 "Samuel", "Road 4.1"
 "Kenneth", "Road 2.6"

Bina satu metod utama untuk memasukkan semua maklumat penumpang ke dalam sistem anda. Selepas itu, sistem anda akan memaparkan output yang berikut:

Passenger Drop-off List
 Stop A: Susan
 Stop A: Benjamin
 Stop B: Kenneth
 Stop D: Rachel
 Stop D: Samuel
 Stop E: Jack

- ◆ Dalam soalan ini, anda dibenarkan untuk mengguna kelas *PriorityQueue* daripada Java API Library.
- ◆ Tidak perlu mengikut urutan untuk penurunan penumpang bagi perhentian yang sama.

You are tasked to build a drop-off system for the automated bus transportation in a smart city. Every workday, at 7 am, the automated bus will pick up a group of passengers from the pick-up point inside the smart city. Upon entry, the automated bus will retrieve passenger's information to arrange for the drop-off sequence.

The retrieved passenger information indicates the passenger's name and the location of the passenger's workplace, for instance, {"Ally", "Road 3.5"} specifies that the passenger's name is Ally and her workplace is located at 3.5 KM from the entrance of the smart city.

The pick-up point of the smart city is at Road 0.2 (viz., 0.2 KM from the entrance of the smart city). The automated bus will stop at every interval of 1.0 KM, starting from the pick-up point. There is a total of 5 bus-stops inside the smart city (excluding the pick-up point), namely Stop A, B, C, D, and E. The automated bus will drop-off each passenger at the bus-stop nearest to their workplace.

By using PriorityQueue and a custom comparator, create a drop-off system to calculate and display the respective passengers at every bus-stop, if the following passenger information are given:

"Susan", "Road 0.9"
 "Jack", "Road 5.5"
 "Benjamin", "Road 1.6"
 "Rachel", "Road 3.8"
 "Samuel", "Road 4.1"
 "Kenneth", "Road 2.6"

Create a main method to insert all passenger information into your system. Then, your system should output the following:

```
Passenger Drop-off List
Stop A: Susan
Stop A: Benjamin
Stop B: Kenneth
Stop D: Rachel
Stop D: Samuel
Stop E: Jack
```

- ◆ In this question, you are allowed to use the PriorityQueue class from the Java API Library.
- ◆ The drop-off sequence of passenger(s) of same stop does not have any order.

(10 markah / marks)

Soalan / Question 5

Taip jawapan untuk soalan ini dalam borang yang diberi.

Type the answers to this question in the form provided.

Fungsi permutasi, $P(n,r)$ ditakrif sebagai $n!/r!$, di mana n dan r ialah dua nombor bulat positif, manakala $!$ mewakili faktorial. Kod di bawah mengira permutasi n dan r secara berulang. Dengan menjawab soalan (a), (b) dan (c) seperti yang ditunjukkan dalam bahagian komen, jadikan kod ini berjalan tanpa kesilapan.

The permutation function, $P(n, r)$ is defined as $n!/r!$, where n and r are two positive integers, whereas $!$ represents factorial. The following code computes the permutation of n and r recursively. By answering questions (a), (b) and (c) as indicated in the comment section, make the code runs without error.

(3 markah / marks)

```

1 // This method receives n and r as positive integers
2 public static int permut(int n, int r) {
3     if (_____) // question (a)
4         return -1;
5     if (n-1>r)
6         _____ // question (b)
7     else
8         _____ // question (c)
9 }

```

Soalan / Question 6

Untuk menjawab soalan ini, serahkan fail `.java` sahaja.

To answer this question, submit the `.java` files only.

Satu sistem pengurusan gudang dilaksanakan dalam sebuah syarikat logistik. Sistem ini dapat menyimpan maksimum 1000 bungkusan, dan setiap bungkusan ditanda dengan teg bungkusan yang bernombor dari 0 hingga 999. Ketika bungkusan sampai di gudang, ia diberi satu teg bungkusan secara rawak berdasarkan ketersediaan teg.

Untuk mencari bungkusan di dalam gudang untuk tujuan penghantaran, satu sistem penglihatan komputer digunakan untuk mencari teg yang dikehendaki dari sebaris bungkusan yang bersusunan rawak. Untuk mengenal pasti nombor atas setiap tag, sistem penglihatan komputer ini memerlukan 15 saat. Anggapkan masa yang diperlukan untuk bergerak dari satu bungkusan ke bungkusan seterusnya adalah kecil dan boleh diabaikan. Sistem ini membuat carian bungkusan secara jujukan sehingga bungkusan dijumpai.

A warehouse management system is implemented for a logistic company. The system can store a maximum of 1000 parcels, each with a parcel tag numbered from 0 to 999. When a parcel reaches the warehouse, it is given a parcel tag with a random number based on the availability of the tags.

To locate a parcel within the warehouse for delivery purpose, a computer vision system is used to search for the requested tag from a line of randomly arranged parcels. To identify the number on each parcel tag, this computer vision requires 15 seconds. Assume that the time required to move from a parcel to next parcel is small enough to be ignored. The system do the parcel search sequentially until the parcel is found.

- a) Laksanakan satu kaedah carian linear yang mencari teg secara jujukan dalam sistem pengurusan gudang. Kaedah tersebut harus mencetak kedudukan bungkusan, dan mengembalikan jumlah masa (dalam minit) yang digunakan untuk pencarian ini.

Implement a linear search method that search for tags sequentially in the warehouse management system. This method should print out the position of the parcel, and return the amount of time (in minutes) spent for the search.

(2 markah / marks)

Untuk mengoptimumkan kecekapan pencarian gudang, bungkusan boleh disusun mengikut turutan secara tertib menaik, disertai dengan penggunaan kaedah pencarian binari. Akan tetapi, sistem lengan robot akan mengambil masa 6 saat untuk melakukan setiap pertukaran bungkusan.

In order to optimize the efficiency of warehouse search, the parcels can be sorted to follow an ascending order of tags, along with the implementation of binary search. However, it will take 6 seconds for a robotic arm system to perform each parcel swap.

- b) Laksanakan kaedah carian binari yang dapat mencari dengan lebih cekap tetapi memerlukan bungkusan disusun mengikut tertib menaik. Kaedah tersebut juga perlu mencetak kedudukan bungkusan, dan mengembalikan jumlah masa (dalam minit) yang digunakan untuk pencarian ini. Untuk berbuat demikian, laksanakan 2 algoritma penyusunan pilihan anda dan sertakan kos (dalam minit) untuk setiap pertukaran bungkusan. Sistem penglihatan komputer yang sama digunakan untuk mengenal pasti nombor atas setiap teg bungkusan.

Implement the binary search method that can search more efficiently but need the parcels to be sorted in an ascending order. The method should also print out the position of the parcel, and return the amount of time (in minutes) needed to retrieve the parcel. In order to do this, implement 2 possible sorting algorithms of your choice and include the cost (in minutes) for each parcel swap. The same computer vision system is used to identify the number on each parcel tag.

(4 markah / marks)

- c) Tulis satu program untuk membandingkan kecekapan kedua-dua sistem semasa gudang adalah 50% penuh. Sistem mana yang lebih cekap (perlu masa yang lebih pendek untuk mencari satu bungkusan)?

Write a program to compare the efficiency of both systems when the warehouse is 50% full. Which system is more efficient (needs shorter time to locate a parcel)?

(3 markah / marks)

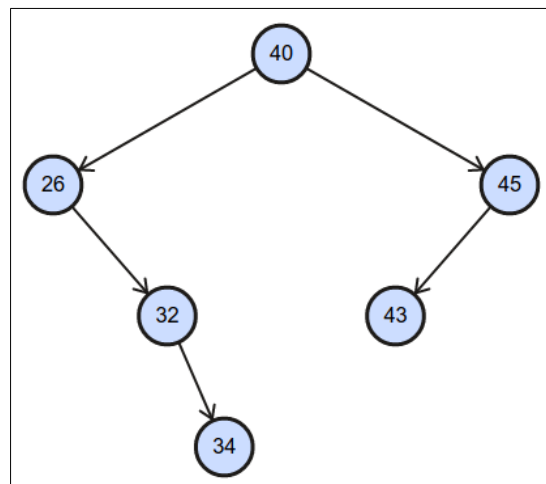
Soalan / Question 7

Untuk menjawab soalan (a), lukis jawapan anda atas kertas, ambil foto dan hantar fail imej. Untuk menjawab soalan (b), taip jawapan anda ke ruang yang disediakan.

To answer question (a), draw the answer on a paper, take photo and submit the image file. To answer question (b), type your answer to the field provided.

Diberi satu Pokok Carian Binari seperti ditunjuk dalam Gambarajah 3.

Given a Binary Search Tree as shown in Figure 3.



Gambarajah / Figure 3

- a) Satu list sebanyak 3 nod dengan nilai-nilai berikut dimasukkan ke pokok mengikut tertib. Lukis satu pokok selepas setiap operasi masukan.

A list of 3 nodes with the following values are inserted to the tree in order. Draw a tree after each insert operation.

41, 38, 18

(3 markah / marks)

- b) Barikan lintasan “preorder”, “inorder” dan “postorder” pokok ini selepas ketiga-tiga nod dimasukkan.

Give the preorder, inorder and postorder traversal of this tree after all the three nodes are inserted.

(3 markah / marks)

Soalan / Question 8

Taip jawapan untuk soalan ini dalam borang yang diberi.

Type the answers of this question in the form provided.

- a) Berikut adalah implimentasi satu graf tidak terarah yang dipermudahkan dengan menggunakan konsep matriks bersebelahan.

The following is a simplified implementation of an undirected graph using the concept of adjacent matrix.

```

1 public class GraphMatrix {
2     int vertex;
3     boolean edges[][];
4
5     public GraphMatrix(int v) {
6         this.vertex = v;
7         edges = new boolean[v][v];
8     }
9
10    public void addEdge(int from, int to) {
11        edges[from][to]=true;
12    }
13
14
15    public void print() {
16        System.out.println("Print the graph:");
17        for (int i = 0; i < vertex; i++) {
18            System.out.print( i + " -> ");
19            for (int j = i; j < vertex ; j++) {
20                if(edges[i][j])
21                    System.out.print(j + " ");
22            }
23            System.out.println("End of the graph:");
24        }
25    }
26 }

```

- i. Terdapat satu penyata yang hilang di baris 12. Tulis penyata tersebut.

There is a missing statement in line 12. Write the statement.

- ii. Terdapat satu kesilapan dalam kaedah *print*. Pastikan nombor baris kesilapan ini.

There is a mistake in the print method. Identify the line number of the mistake.

(2 markah / marks)

- b) Terdapat satu algoritma misteri yang ditakrif seperti berikut. Algoritma ini membaca satu graf *G* dan satu bucu *v* dalam grah tersebut, dan menghasilkan satu pokok. Kenalpastikan nama algoritma ini.

*There is an mysterious algorithm defined as follow. This algorithm reads a graph *G* and a vertex *v* in the graph, and produce a tree *T*. Identify the name of this algorithm.*

```

Mark vertex  $v$  as visited.
for each of  $v$ 's outgoing edges,  $e = (v, w)$  do
    if vertex  $w$  has not been visited then
        Record edge  $e$  as the discovery edge for vertex  $w$ .
        Recursively call back itself with graph  $G$  and vertex  $w$ .

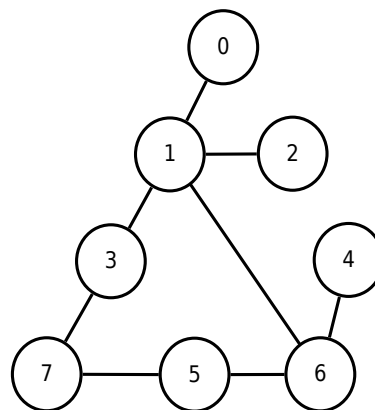
```

(1 markah / mark)

- c) Satu graf ditunjukkan dalam Gambarajah 4 dihantar untuk diproses oleh algoritma dalam 5(b). Jika akar pokok yang dibentuk ialah 0, kenal pasti:

A graph shown in Figure 4 is sent to the algorithm in 5(b) to process. If the root of the tree formed is 0, identify:

- i. Ayah kepada 5.
The parent of 5.
- ii. Adik-beradik kepada 3.
The sibling of 3.
- iii. Anak kepada 6.
The child of 6.
- iv. Bucu terakhir ditanda sebagai dilawati.
The last vertex being marked as visited.



Gambarajah / Figure 4

(4 markah / marks)

TAMAT
END