

UNIVERSITI MALAYA
UNIVERSITI MALAYA

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

SESI AKADEMIK 2022/2023
ACADEMIC SESSION 2022/2023

: SEMESTER 2
: SEMESTER 2

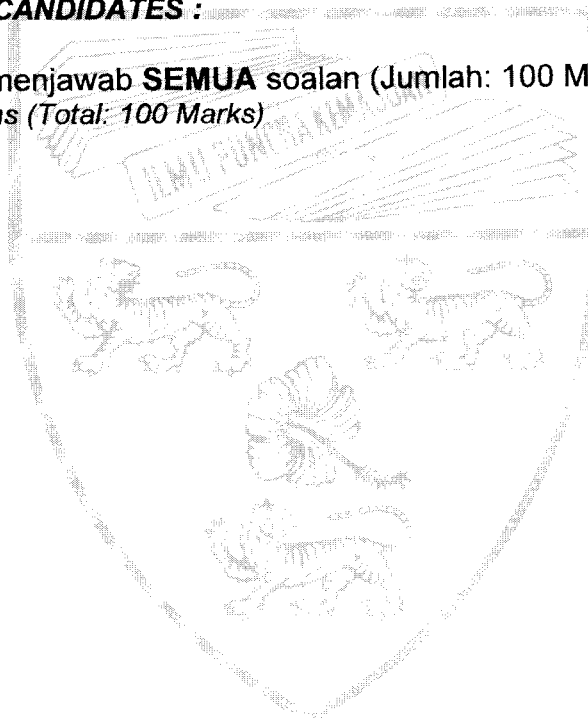
WIA1002 : Struktur Data
Data Structure

Julai 2023
July 2023

MASA : 3 jam
TIME : 3 hours

ARAHAN KEPADA CALON :
INSTRUCTIONS TO CANDIDATES :

Calon dikehendaki menjawab **SEMUA** soalan (Jumlah: 100 Markah)
Answer **ALL** Questions (Total: 100 Marks)



(Kertas soalan ini mengandungi 6 soalan dalam 15 halaman yang dicetak)
(This question paper consists of 6 questions on 15 printed pages)

1. Berdasarkan perpustakaan Java `java.util.LinkedList`, bangunkan `ExamStack`, satu kelas timbunan generik. Kelas ini perlu ada 4 kaedah sahaja, iaitu `peek`, `pop`, `push` dan `getSize`.

Based on the Java library `java.util.LinkedList`, develop `ExamStack`, a generic stack class. This class should have 4 methods only, i.e. `peek`, `pop`, `push` and `getSize`.

[3 markah/marks]

2. Aritmetik merupakan kajian asas dalam matematik yang melibatkan operan dan operator. Operasi asas yang melibatkan penambahan, penolakan, pendaraban dan pembahagian diajar dalam pendidikan rendah untuk memastikan pelajar membina asas yang kukuh untuk topik lain dalam matematik. Untuk membantu pelajar dalam pendidikan rendah, anda diminta oleh guru mereka untuk menulis program yang membaca ungkapan aritmetik dari input pengguna dan menghasilkan jawapan serta-merta. Anda dikehendaki menggunakan hanya kelas `ExamStack` yang dibangunkan di Soalan 1 untuk menyelesaikan masalah ini dan tiada struktur data lain dibenarkan.

Anda juga dikehendaki menilai ungkapan seperti yang diberikan. Jangan tukarkannya kepada Notasi Poland, Notasi Poland Songsang atau bentuk lain untuk penilaian. Memandangkan program ini dibangunkan untuk pendidikan rendah, anda hanya perlu menilai penambahan, penolakan, pendaraban, pembahagian dan kurungan. Tambahan pula, semua operan adalah integer positif satu digit tetapi output boleh menjadi nombor nyata. Anda boleh menganggap input daripada pengguna sentiasa merupakan ungkapan aritmetik yang sah, dan pengguna tidak akan memberikan ungkapan yang melibatkan pembahagian dengan 0.

Contoh input dan output program diberikan di bawah:

Arithmetic is an elementary study in mathematics which involves operands and operators. Fundamental operations involving addition, subtraction, multiplication and division are taught in primary education to ensure students build a strong foundation for other topics in mathematics. To assist students in primary education, you are asked by their teachers to write a program that reads arithmetic expressions from user inputs and produces answers immediately. You are required to use only the `ExamStack` class developed in Question 1 to solve this problem, and no other data structures are allowed.

You are also required to evaluate the expressions as given, do not convert them to Polish Notations, Reverse Polish Notations or other forms for evaluations. Since the program is developed for primary education, you only need to evaluate addition, subtraction, multiplication, division and braces. Furthermore, all the operands are single digit positive integers but the output can be real numbers. You can assume the inputs from the users are always valid arithmetic expressions, and users will not provide expressions that involve division by 0.

Sample inputs and outputs of the program are given below:

Enter expression to evaluate : 5-3+4 5-3+4 = 6.0
Enter expression to evaluate : 2+4*5-(9+7*6) 2+4*5-(9+7*6) = -29.0
Enter expression to evaluate : 9-2*(5/2-3/5+2)/6 9-2*(5/2-3/5+2)/6 = 7.7

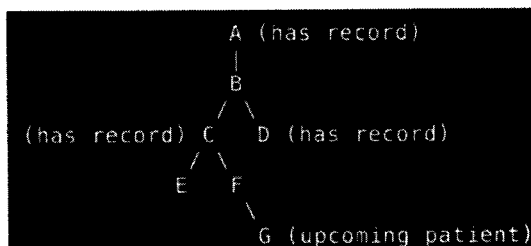
[12 markah/marks]

3. Buku keluarga (**family_book.csv**) ialah satu jadual yang mengandungi nama semua orang di bandar, maklumat asas mereka dan ibu bapa mereka. Dan anda adalah seorang doktor yang akan berjumpa dengan ramai pesakit setiap minggu. Di hospital anda, anda mempunyai satu rekod pesakit yang lalu dan maklumat perubatan mereka (**past_medical_record.csv**). Dengan merujuk kepada buku keluarga dan rekod lampau, anda ingin memprofilkan pesakit anda yang akan datang (**upcoming_patients.csv**) dengan melihat rekod perubatan merujuk kepada nenek moyang terdekat mereka (ibu bapa / ibu bapa ibu bapa, dll) sekiranya terdapat kemungkinan penyakit keturunan.

A family book (**family_book.csv**) is a table that contains the names of all people in the city, their basic information and their direct parent. And you are a doctor who will see many patients every week. In your hospital, you have a record of past patients and their medical information (**past_medical_record.csv**). By referring to the family book and the past records, you would like to profile your upcoming patients (**upcoming_patients.csv**) by looking at an existing medical record of their closest ancestor (parent / parent of the parent, etc.) in case there is a possibility of hereditary disease.

Contoh keluarga ditunjukkan di bawah.

An example of the family tree is shown below.



Untuk pesakit **G** yang akan datang, walaupun **A**, **C** dan **D** (nenek moyang/saudara mara) mempunyai rekod perubatan, kami hanya berminat dengan nenek moyang terdekat iaitu **C**. Selain itu, untuk pesakit **G** yang akan datang, walaupun **F** adalah ibu bapanya, tapi tidak ada rekod perubatan masa lalu untuk **F**. Oleh itu, kita perlu terus mencari keturunan pesakit **G** dengan mencari moyang yang terdekat bersama rekod perubatan, iaitu **C**. Memandangkan **A** berada di bahagian atas salasilah keluarga, anda boleh menganggap bahawa orang itu berhijrah ke bandar pada masa lalu dan oleh itu tidak mempunyai maklumat ibu bapa dalam buku keluarga.

Andaian

- a) tidak boleh ada dua orang yang mempunyai nama yang sama
- b) setiap orang hanya boleh mempunyai seorang ibu bapa yang disenaraikan dalam buku keluarga
- c) seorang ibu bapa boleh mempunyai beberapa anak
- d) tidak mempunyai maklumat ibu bapa dalam buku keluarga bermakna orang itu berhijrah ke bandar (bahagian atas salasilah keluarga)
- e) mempunyai nama keluarga yang sama tidak menjamin dua orang atau lebih tergolong dalam keluarga yang sama

*For upcoming patient **G**, even though **A**, **C** and **D** (ancestors/relatives) have medical record, we are only interested in the closest ancestor which is **C**. Additionally, for upcoming patient **G**, although **F** is its direct parent, there are no past medical records for **F**. Therefore, we need to keep searching patient **G**'s ancestry to find the closest ancestor with a medical record, which is **C**. Since **A** is at the top of the family tree, you can assume that the person migrated into the city in the past and thus have no parent information in the family book.*

Assumptions

- a) *there can be no two people with the same name*
- b) *each person can only have one parent listed in the family book*
- c) *one parent can have multiple children*
- d) *having no parent information in the family book means the person migrated into the city (top of the family tree)*
- e) *having the same last name does not guarantee two or more person belongs to the same family*

Contoh Jadual

Semua fail **csv** dipisahkan koma dengan baris baharu yang memisahkan setiap baris. Baris pertama setiap fail sentiasa menjadi pengepala jadual.

- Buku Keluarga

Examples of Tables

All **csv** files are comma separated with newline separating each rows. The first line of each file is always the table header.

- Family Book

FirstName	LastName	ParentFirstName	ParentLastName
Morgana	Chris		
Andree	Alica		
Malinde	Giffie	Lucia	Giffie

Rujuk kepada andaian 4 di mana `ParentFirstName` dan `ParentLastName` adalah kosong.

Refer to assumption 4 where `ParentFirstName` and `ParentLastName` are empty.

- Rekod Perubatan Lepas

Past Medical Record

FirstName	LastName	Height	Weight	Occupation	Diabetic	HeartDisease	Smoking	DrinkAlcohol
Andre	Ebert	153	147	Coil Winder	Yes	No	Yes	No
Faye	Nora	151	68	Animator	No	No	No	Yes

- Pesakit Akan Datang

Upcoming Patients

FirstName	LastName
Rosana	Blain
Rheba	Doersten

Output yang Dijangka

Anda dikehendaki mencipta satu fungsi dengan hujah `wildcard` untuk memaparkan sebarang jenis jadual (fail `csv` atau jawapan akhir anda) dalam output sampel.

Sila gunakan fungsi anda di atas untuk memaparkan jawapan akhir anda - jadual dengan semua nama pesakit yang akan datang, dan nama serta rekod perubatan moyang terdekat mereka (awalan "Ibu bapa" dalam pengepala jadual).

Apabila tiada rekod perubatan ditemui dalam keturunan pesakit yang akan datang, lajur hanya diisi dengan "Tiada".

Expected Output

You are required to create a function with `wildcard` arguments to display any type of table (the `csv` files or your final answer) in the sample output.

Please use your function above to display your final answer - a table with all upcoming patient names, and the names and medical records of their closest ancestor (prefixed "Parent" in the table header).

When no medical records are found in an upcoming patient's ancestry, the columns are simply filled with "None".

Keluaran sampel

Sample output

FirstName	LastName	ParentFirstName	ParentLastName	ParentHeight	ParentWeight	ParentOccupation	ParentDiabetic	ParentHeartDisease	ParentStroke	ParentDrunkAlcohol
Justin	Bieber	Margaret	Bieber	168	70	Office worker	Yes	Yes	No	Stopped
Rebecca	Black	None	None	None	None	None	None	None	None	None

Ambil perhatian bahawa lajur dipisahkan oleh tab dan baris dipisahkan oleh baris baharu. Anda boleh mulakan dengan contoh kod dalam `Main.java`

Note that columns are separated by tabs and rows are separated by newline. You can start with the example codes in `Main.java`

[25 markah/marks]

4. Stesen cuaca automatik (AWS) ialah stesen cuaca moden untuk mengumpul ukuran dari kawasan terpencil. AWS biasanya terdiri daripada kandang kalis cuaca yang mengandungi pencatat data, bateri boleh dicas semula, penerima meteorologi dan dipasang pada tiang. AWS menawarkan beberapa kelebihan termasuk pelaporan automatik pada resolusi temporal yang sangat halus. Data yang dikumpul oleh sistem AWS yang berbeza adalah dalam format yang berbeza.

Fail `IOES_Bachok_14093367_03122014.elab` adalah diekstrak daripada sistem LSI LASTEM Gidas yang digunakan oleh Institut Sains Lautan dan Bumi (IOES) UM di Stesen Penyelidikan Marin Bachok. Fail elab terdiri daripada 28 atribut, termasuk kelembapan relatif, suhu udara, tekanan atmosfera, kelajuan angin, arah angin, bekalan kuasa kepada sistem AWS, suhu dalaman, titik embun dan yang paling penting, masa pengukuran ini telah diambil.

Tugas anda ialah menulis satu program untuk membantu ahli oseanografi mengekstrak dan menganalisis data AWS. Program anda harus termasuk fungsi yang disenaraikan di bawah:

An automatic weather station (AWS) is a modern weather station to collect measurements from remote areas. An AWS will typically consist of a weather-proof enclosure containing the data logger, rechargeable battery, meteorological sensors and mounted upon a mast. AWS offers several advantages including automated reporting at a very fine temporal resolution. The data collected by different AWS systems are in different formats.

The given `IOES_Bachok_14093367.03122014.elab` file is extracted from the LSI LASTEM Gidas system deployed by the UM Institute of Ocean and Earth Sciences (IOES) in the Bachok Marine Research Station. The `elab` file consists of 28 attributes, which include the relative humidity, air temperature, atmospheric pressure, wind speed, wind direction, power supply to the AWS system, internal temperature, dew point and most importantly, the time when these measurements have been taken.

Your task is to write a program to help oceanographers extract and analyse AWS's data. Your program should include the functionalities listed below:

- (a) Baca masa pengukuran (Tarikh/masa), kelembapan relatif (RELHumidity – Ave), suhu udara (AIRTemp – Ave), tekanan atmosfera (ATMPressure – Ave) kelajuan angin (WindSPEED – Ave) dan arah angin (WindDIR – RisDir) ke dalam struktur data, dan cetak kandungan pada skrin sebagai jadual enam lajur.

Read the measurement time (Date/time), relative humidity (RELHumidity – Ave), air temperature (AIRTemp – Ave), atmospheric pressure (ATMPressure – Ave) wind speed (WindSPEED – Ave) and wind direction (WindDIR – RisDir) into a data structure, and print the content on the screen as a six columns table.

- (b) Keluarkan hingar (iaitu anomali atau outlier yang disebabkan oleh sensor rosak atau hanyut) menggunakan *Moving Median*, dengan saiz tetingkap 3.

[Petunjuk]: Berdasarkan jadual di bawah, nilai ukuran ketiga 2, ialah nilai median bagi tiga ukuran pertama [1,2,99].

Remove the noise (i.e. anomalies or outliers caused by sensors faulty or drift) using *Moving Median*, with window size 3.

[Hint]: Based on table below, the value of the third measurement 2, is the median value of the first three measurements [1,2,99].

Timestamp	Before denoise	After denoise
2014-12-03T02:50:00	1	N/A
2014-12-03T03:00:00	2	N/A
2014-12-03T03:10:00	99	2
2014-12-03T03:20:00	4	4
2014-12-03T03:30:00	5	5
2014-12-03T03:40:00	6	5
2014-12-03T03:50:00	55	6
2014-12-03T04:00:00	8	8
2014-12-03T04:10:00	9	9
2014-12-03T04:20:00	10	9

- (c) Tukar arah mata angin dari darjah kepada arah mata angin berdasarkan jadual di bawah. Sebagai contoh, jika arah angin ialah 24.52 darjah, ia perlu ditukar kepada NNE. Untuk nilai bertindih, arah utama harus digunakan. Sebagai contoh, 11.25 darjah sepatutnya N, bukannya NNE.

Convert the wind direction from degree to cardinal, based on the table below. As an example, if the wind direction is 24.52 degrees, it should be converted to NNE. For overlapping values, the main direction should be used. For instance, 11.25 degree should be N, instead of NNE.

Cardinal Direction	Degree Direction
N	348.75 - 11.25
NNE	11.25 - 33.75
NE	33.75 - 56.25
ENE	56.25 - 78.75
E	78.75 - 101.25
ESE	101.25 - 123.75
SE	123.75 - 146.25
SSE	146.25 - 168.75
S	168.75 - 191.25
SSW	191.25 - 213.75
SW	213.75 - 236.25
WSW	236.25 - 258.75
W	258.75 - 281.25
WNW	281.25 - 303.75
NW	303.75 - 326.25
NNW	326.25 - 348.75

- (d) Kira Tekanan Wap Tepu menggunakan persamaan Antoine di bawah dengan pemalar untuk air:

$$\log_{10} P = A - \frac{B}{C + T}$$

di mana T ialah suhu dalam Kelvin (K) dan tekanan wap P adalah dalam mmHg. Pemalar diberikan sebagai:

A	B	C	$T_{\min}, ^\circ\text{C}$	$T_{\max}, ^\circ\text{C}$
8.07131	1730.63	233.426	1	99
8.14019	1810.94	244.485	100	374

Calculate the Saturated Vapor Pressure using the Antoine equation below with the constants for water:

$$\log_{10} P = A - \frac{B}{C + T}$$

where T is temperature in Kelvin (K) and the vapor pressure P is in mmHg. The constants are given as:

A	B	C	$T_{\min}, ^\circ\text{C}$	$T_{\max}, ^\circ\text{C}$
8.07131	1730.63	233.426	1	99
8.14019	1810.94	244.485	100	374

[Petunjuk]: Suhu yang direkodkan oleh sistem AWS adalah dalam darjah Celsius ($^\circ\text{C}$). Anda harus menukarnya kepada Kelvin dengan menambah 273.15.

[Hint]: The temperature recorded by the AWS system is in degrees Celsius ($^\circ\text{C}$). You should convert it into Kelvin by adding 273.15.

- (e) Akhir sekali, cetak kandungan struktur data anda, selepas melaksanakan tugas (b) hingga (d), dengan tekanan wap tepu ditambah sebagai lajur ke-7 dalam jadual anda.

Lastly, print the content of your data structure, after performing task (b) to (d), with the saturated vapor pressure added as the 7th column in your table.

[20 markah/marks]

5. Anda diberi senarai nama yang mengandungi kedua-dua nama Inggeris dan Rusia. Tugas anda ialah untuk i) mengisih senarai nama mengikut susunan leksikografi tertentu, ii) mencari indeks nama tertentu yang diberikan dalam senarai yang diisih, dan iii) mencari nama dengan huruf terbanyak dalam senarai. Beberapa huruf Rusia boleh ditulis menggunakan huruf Inggeris yang setara, jadi anda mesti mengisih senarai menggunakan susunan abjad tertentu yang disediakan di bawah:

You are given a list of names that contains both English and Russian names. Your task is to i) sort the name list according to certain lexicographical order, ii) find the index of specific given names in the sorted list, and iii) find the name with the most letters in the list. Some Russian letters can be written using English letter equivalents, so you must sort the list using a specific alphabetical order that is provided below:

<i>i</i>	EN	RU	<i>i</i>	EN	RU	<i>i</i>	EN	RU	<i>i</i>	EN	RU
0	A	А	9	G	Г	18	N	Н	27	U	У
1	B	Б	10	H	Х	19	O	О	28	V	В
2	C	Ц	11	I	И	20	P	П	29	W	
3		Ч	12	J	Й	21	Q		30	X	
4	D	Д	13		Я	22	R	Р	31	Y	Ы
5	E	Е	14		Ю	23	S	С	32		Ж
6		Ё	15	K	К	24		Ш	33	Z	З
7		Э	16	L	Л	25		Щ	34		Ь
8	F	Ф	17	M	М	26	T	Т	35		б

Jika dua nama mempunyai tahap keutamaan yang sama hendaklah diisih berdasarkan susunan ia muncul dalam senarai yang diberikan. Sebagai contoh, kedua-dua MARIA (EN) dan МАРИА (RU) mempunyai tahap keutamaan yang sama dan harus diisih berdasarkan siapa yang muncul dahulu dalam senarai yang diberikan. Anda boleh mengandaikan bahawa

semua nama yang diberikan akan BERHUBUNG MODAL. Semua nama yang diberikan dalam latihan ini akan mempunyai sekurang-kurangnya satu huruf, dan ia hanya akan mengandungi aksara yang disenaraikan dalam jadual yang disediakan di atas. Sila berhati-hati bahawa sesetengah aksara mungkin kelihatan serupa tetapi mempunyai kod bait yang berbeza: A (\u0041), A (\u0410).

Untuk memudahkan anda, senarai lengkap huruf dalam susunan yang ditentukan dibentangkan dalam satu baris yang ditunjukkan di bawah dan mana-mana ruang kosong diwakili menggunakan ".".

*If two names have the same precedence level should be sorted based on the order they appear in the given list. For instance, both MARIA (EN) and МАРИА (RU) have the same precedence level and should be sorted based on who appears first in the given list. You can safely assume that all names given will be **CAPITALIZED**. All the names given in this exercise will have at least one letter and will only contain the characters listed in the table above. Please be very careful that some characters may appear similar but have different byte codes: A (\u0041), A (\u0410).*

To make it easier, the complete list of letters in the specified order is presented in a single line shown below, and any empty spaces are represented using ".".

EN: ABC.DE.FGHIJ.KLMNOPQRS.TUVWXY.Z.

RU: АБЦЧДЕЁЭФГХИЙЯЮКЛМНОП.РСШЩТУВ..ЫЖЗЬБ

- (a) Untuk tugas ini, anda boleh memulakan kerja anda dengan kaedah main yang disediakan di bawah:

For this task, you can begin your work with the `main` method provided below:

```
public static void main(String[] args) {
    String[] names = {
        "DAVID", "ELIZABETH", "JAMES", "JENNIFER", "JOHN",
        "LINDA", "MARIA", "MICHAEL", "PATRICIA", "ROBERT",
        "АНТОНОВ", "ВОЛКОВ", "ГУРИН", "ИВАНОВ", "КОЛМОГОРОВ",
        "МАРИА", "МАРКОВ", "ПЕТРОВ", "ПУТИН", "СТЕПАНОВ"
    };

    Comparator<String> comparator = new SpecialComparator<>();
    sort(names, comparator);
    System.out.print("# Sorted name list:\n - ");
    printArray(names, "\n - ");
}
```

Seperti yang mungkin anda perhatikan daripada kod yang diberikan, tugas anda adalah untuk mencipta `Comparator` generik yang dipanggil `SpecialComparator` dengan parameter jenis yang **dibatasi atas** oleh `CharSequence`. Pembandingan ini akan menentukan susunan dua nama

yang diberi. Secara khususnya, kaedah bandingkan akan mengembalikan nilai negatif jika nama pertama harus didahulukan sebelum nama kedua, sifar jika kedua-dua nama mempunyai kepentingan yang sama, atau nilai positif jika nama pertama harus datang selepas nama kedua.

Seterusnya, anda perlu menulis kaedah isihan generik yang mengambil tatasusunan jenis generik dan objek Pembandingan untuk perbandingan. Anda bebas menggunakan sebarang jenis algoritma pengisihan, termasuk isihan gelembung yang remeh.

Akhir sekali, anda juga harus menulis kaedah `printArray` generik yang mencetak setiap elemen dalam tatasusunan generik diikuti dengan pembatas yang diberikan. Anda seharusnya melihat output berikut jika pelaksanaan anda betul:

*As you may have noticed from the code provided, your task is to create a generic `Comparator` called `SpecialComparator` with a type parameter that is **upper bounded** by `CharSequence`. This comparator will determine the order of two given names. Specifically, the `compare` method will return a negative value if the first name should come before the second name, zero if both names have the same importance, or a positive value if the first name should come after the second name.*

Next, you'll need to write a generic `sort` method that takes an array of generic type and a `Comparator` object for the comparisons. You're free to use any type of sorting algorithm, including the trivial bubble sort.

Finally, you should also write a generic `printArray` method that print each element in a generic array followed by the given `delimiter`. You should see the following output if your implementation is correct:

```
# Sorted name list:
- АНТОНОВ
- DAVID
- ELIZABETH
- ГУРИН
- ИВАНОВ
- JAMES
- JENNIFER
- JOHN
- КОЛМОГОРОВ
- LINDA
- MARIA
- МАРИА
- МАРКОВ
- MICHAEL
- PATRICIA
- ПЕТРОВ
- ПУТИН
- ROBERT
- СТЕПАНОВ
- ВОЛКОВ
```

- (b) Untuk tugas ini, anda perlu menulis carian kaedah generik yang boleh mencari indeks item tertentu dalam tatasusunan. Output yang dijangkakan untuk tugas ini ditunjukkan di bawah:

For this task, you need to write a generic method `search` that can locate the index of a given item within an array. The expected output for this task is shown below:

```
# Search
Index of АНТОНОВ: 0
Index of MARKOV: -1
Index of МАРИА: 11
Index of ВОЛКОВ: 19
```

- (c) Tugas akhir anda ialah mencari nama terpanjang dalam senarai. Jika dua atau lebih nama mempunyai panjang yang sama, maka nama yang muncul dahulu dalam senarai harus dianggap sebagai jawapannya. Untuk mencapai matlamat ini, anda perlu mencipta kaedah generik yang dipanggil `findMax`. Kaedah ini harus mengambil tatasusunan generik dan `Comparator` sebagai parameter, dan mengembalikan indeks elemen maksimum dalam tatasusunan berdasarkan hasil perbandingan daripada Perbandingan.

Kemudian, anda perlu mencipta `Comparator` yang membandingkan panjang dua rentetan yang diberikan. Anda boleh melakukan ini dengan

sama ada mencipta objek `Comparator` baharu atau membuat instantiation objek `Comparator` menggunakan ungkapan kelas tanpa nama.

Untuk makluman anda, nama terpanjang dalam senarai ialah "КОЛМОГОРОВ", dan indeksnya dalam senarai diisih ialah 8.

[PETUNJUK]: `Comparator` ialah antara muka Java yang digunakan untuk menentukan jumlah susunan objek. Ia mempunyai dua kaedah `int compare(T o1, T o2)` dan `boolean equals(Object obj)`, tetapi kita hanya perlu mentakrifkan kaedah `compare`. Anda boleh menentukan logik perbandingan tersuai anda sendiri untuk mengisi objek dalam susunan tertentu.

Your final task is to find the longest name in the list. If two or more names are the same length, the name that appears first in the list should be considered the answer. To achieve this, you need to create a generic method called `findMax`. This method should take a generic array and a `Comparator` as parameters, and return the index of the maximum element in the array based on the comparison results from the `Comparator`.

Then, you will need to create a `Comparator` that compares the length of two given strings. You can do this by either creating a new `Comparator` object or instantiating a `Comparator` object using an anonymous class expression.

For your information, the longest name in the list is "КОЛМОГОРОВ", and its index in the sorted list is 8.

[HINTS]: `Comparator` is a Java interface that is used to define the total ordering of the objects. It has two methods `int compare(T o1, T o2)` and `boolean equals(Object obj)`, but we only need to define the `compare` method. You can define your own custom comparison logic to sort objects in a specific order.

Contoh:

Mengisi tatasusunan dalam tertib menurun.

Example:

Sorting an array in descending order.

```

import java.util.Arrays;
import java.util.Comparator;

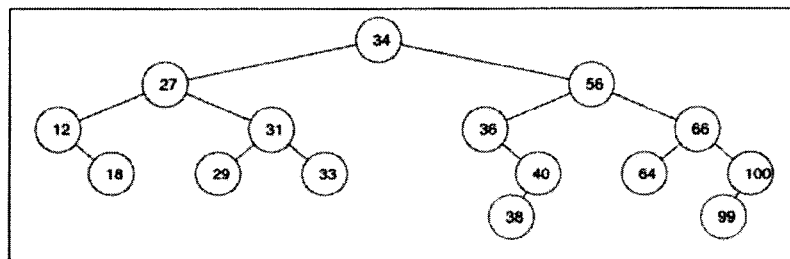
class Example
{
    public static void main(String[] args)
    {
        Integer[] array = {20, 10, 77, 36, 39, 98, 34, 96, 44, 64};
        Arrays.sort(array, new Comparator<Integer>() {
            @Override
            public int compare(Integer o1, Integer o2) {
                return o2 - o1;
            }
        });
        System.out.println(Arrays.toString(array));
        // [98, 96, 77, 64, 44, 39, 36, 34, 20, 10]
    }
}

```

[20 markah/marks]

6. Rajah di bawah menunjukkan satu pokok carian binari.

Figure below shows a binary search tree.



- (a) Tuliskan urutan node-node jika pokok dilintasi dengan lintasan *inorder*, *preorder*, dan *postorder*.

Write the sequence of the nodes if the tree is traversed with *inorder*, *preorder* and *postorder* traversal.

- (b) Dalam senario biasa, penghapusan nod dalam pokok carian binari melibatkan mencari nod paling kanan dalam anak kiri dan menggantikan nod yang dipadam dengan nod paling kanan yang dijumpai, jika anak kiri wujud. Melaksanakan semula penyelesaian untuk penghapusan nod dan gunakan nod paling kiri anak kanan untuk menggantikan nod yang dipadamkan. Bandingkan hasil penyelesaian asal dan penyelesaian yang dilaksanakan semula untuk penghapusan nod "12", "34" dan "66".

In common scenario, deletion of a node in a binary search tree involves looking for the rightmost node of the left child and replace the deleted node with the rightmost node found, if a left child exists. Reimplement the solution for node deletion and use leftmost node of the right child to replace the deleted node instead. Compare the outcome of the original solution and the reimplemented solution for the node deletion of "12", "34" and "66".

- (c) Secara konseptual, pokok carian binari dan senarai pautan dua kali berbulat kelihatan serupa. Kedua-duanya terdiri daripada unsur nod, dan 2 pemboleh ubah yang menunjuk kepada nod lain. Tulis program untuk menukar pokok carian binari ke senarai terpaut dua kali berbulat.

[PETUNJUK]: Konsep rekursif boleh membantu.

Conceptually, a binary search tree and a circular doubly linked list looks similar. Both consist of a node element, and 2 variables that points to another node. Write a program to convert a binary search tree to a circular doubly linked list.

[HINTS]: A recursive concept could help.

[20 markah/marks]

**TAMAT
END**