

SULIT



KEMENTERIAN PENDIDIKAN TINGGI
JABATAN PENDIDIKAN POLITEKNIK DAN KOLEJ KOMUNITI

BAHAGIAN PEPERIKSAAN DAN PENILAIAN
JABATAN PENDIDIKAN POLITEKNIK DAN KOLEJ KOMUNITI
KEMENTERIAN PENDIDIKAN TINGGI

JABATAN TEKNOLOGI MAKLUMAT & KOMUNIKASI

PEPERIKSAAN AKHIR

SESI II : 2024/2025

DFC30233 : DATA STRUCTURES

TARIKH : 15 MEI 2025

MASA : 11.30 PAGI - 1.30 PETANG (2 JAM)

Kertas soalan ini mengandungi **DUA PULUH TIGA (23)** halaman bercetak.

Bahagian A: Objektif (30 soalan)

Bahagian B: Struktur (2 soalan)

Dokumen sokongan yang disertakan : Tiada

JANGAN BUKA KERTAS SOALAN INI SEHINGGA DIARAHKAN

(CLO yang tertera hanya sebagai rujukan)

SULIT

SECTION A : 45 MARKS
BAHAGIAN A : 45 MARKAH

INSTRUCTION:

This section consists of **THIRTY (30)** objective questions. Mark your answers in the OMR form provided.

ARAHAN:

*Bahagian ini mengandungi **TIGA PULUH (30)** soalan objektif. Tandakan jawapan anda di dalam borang OMR yang disediakan.*

CLO1

1. Identify the **CORRECT** answer for non-linear data structure.
*Kenal pasti jawapan yang **BETUL** bagi struktur data bukan linear.*

- A. Queue
Baris Gilir
- B. Array
Tatasusunan
- C. Linked List
Senarai Berpaut
- D. Binary Tree
Pepohon Dedua

CLO1

2. Choose how many cinemaBooking structures can be stored in the GSCIpoh array.
Pilih berapa banyak struktur cinemaBooking boleh disimpan dalam array GSCIpoh.

```
struct cinemaBooking
{
    char MovieTitle[50];
    int duration;
    float price;
} GSCJitra[50], GSCIpoh[75];
```

Figure A2 / Rajah A2

- A. 50
- B. 75
- C. 125
- D. 150

- CLO1 3. Relate the reason arrays are considered inefficient for insertion and deletion.
Kaitkan sebab mengapa tatasusunan dianggap tidak cekap untuk penyisipan dan penghapusan.
- A. They require resizing at runtime.
Ia memerlukan perubahan saiz semasa waktu larian.
 - B. They have fixed memory allocation.
Ia mempunyai peruntukan memori tetap.
 - C. Insertion and deletion require shifting elements.
Penyisipan dan penghapusan memerlukan pemindahan elemen.
 - D. Arrays do not support random access.
Tatasusunan tidak menyokong capaian rawak.
- CLO1 4. Select the real-life application that best represents the use of a Single Linked List.
Pilih aplikasi dunia sebenar yang terbaik menggambarkan penggunaan Senarai Berpaut Tunggal.
- A. Driving direction using Waze.
Navigasi arah memandu menggunakan Waze.
 - B. Music Playlist.
Senarai Lagu Muzik.
 - C. Undo/Redo in Text Editor.
Buat asal / Buat semula dalam Penyunting Teks.
 - D. Lift in a building.
Lif dalam bangunan.
- CLO1 5. Identify a linked list where the last node points to the first node.
Kenal pasti senarai berpaut yang mempunyai nod terakhir yang menunjuk kepada nod pertama.
- A. Singly linked list.
Senarai Berpaut Tunggal.
 - B. Doubly linked list.
Senarai berpaut berganda.
 - C. Circular linked list.
Senarai berpaut membulat.
 - D. Circular doubly linked list.
Senarai berpaut berganda membulat.

CLO1 6. Select the operation that can be performed in List.
Pilih operasi yang boleh dilakukan dalam Senarai.

- i. Insertion.
Penyisipan.
- ii. Deletion.
Penghapusan.
- iii. Traversing.
Penjelajahan.
- iv. Copying.
Menyalin.

- A. i, ii and iii
i, ii dan iii
- B. i, ii and iv
i, ii dan iv
- C. i, iii and iv
i, iii dan iv
- D. i, ii, iii and iv
i, ii, iii dan iv

CLO1 7. Based on Figure A7, identify the last node.
Berdasarkan Rajah A7, kenal pasti nod terakhir.

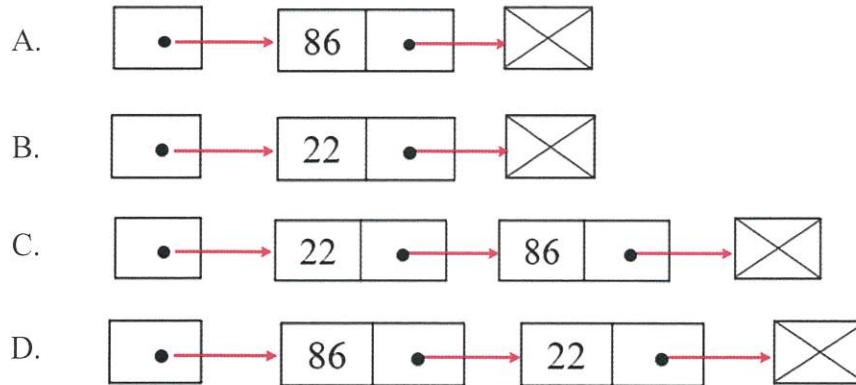


Figure A7 / *Rajah A7*

- A. M
- B. E
- C. Q
- D. J

CLO1

8. Assume that an empty linked list was created. Predict the new linked list after the sequence numbers of **22** and **86** are inserted in the linked list.
Senarai berpaut yang kosong telah dihasilkan. Pilih senarai berpaut baru selepas urutan nombor 22 dan 86 dimasukkan pada senarai berpaut.



CLO1

9. Identify what happens when a stack exceeds its maximum size.
Kenalpasti apa yang berlaku apabila tindanan melebihi saiz maksimumnya.

- A. Underflow Push
- B. Overflow
- C. Deadlock
- D. Segmentation fault

CLO1

10. Select the **CORRECT** real-life scenario that represents the concept of a stack.
*Pilih senario kehidupan sebenar yang **BETUL** yang mewakili konsep stack.*

- A. People waiting to board an elevator enter and exit in the same sequence.
Orang yang menunggu untuk menaiki lif masuk dan keluar dalam urutan yang sama.
- B. Booking systems for events or tickets handle requests in the order they are received.
Sistem tempahan untuk acara atau tiket mengendalikan permintaan mengikut urutan yang diterima.
- C. Customers waiting in a call queue are assisted in the order they called.
Pelanggan yang menunggu dalam talian panggilan dibantu mengikut urutan mereka membuat panggilan..
- D. When using the back button, the most recently visited page is the first to be accessed.
Apabila menggunakan butang kembali, halaman yang paling baru dilawati adalah yang pertama diakses.

CLO1

11. Interpret the most efficient way to check whether a stack is empty in an array-based implementation.

Tafsirkan cara paling berkesan untuk menyemak sama ada tindanan kosong dalam pelaksanaan berasaskan tatasusunan.

A. Check if the array is full.

Periksa jika tatasusunan penuh.

B. Check if the "top" pointer is -1.

Periksa jika penunjuk "top" adalah -1.

C. Check if the array has no elements.

Periksa jika tatasusunan tiada elemen.

D. Check if the stack size is zero.

Periksa jika saiz tumpukan adalah sifar.

CLO1

12. Determine the final output of stack operation in Figure A12 after performing the following operations.

Tentukan keluaran akhir operasi tindakan dalam Rajah A12 selepas melakukan operasi berikut.

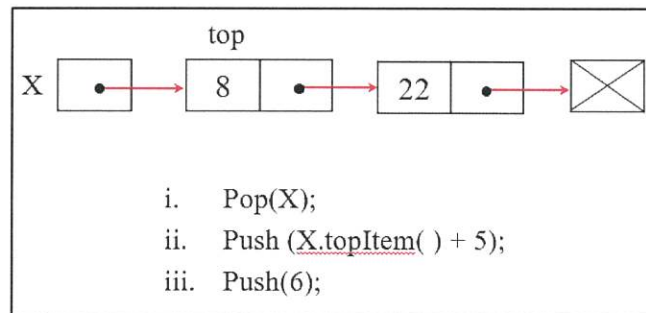


Figure A12 / Rajah A12

- A.
- B.
- C.
- D.

CLO1

13. Assume a stack is implemented using an array with a capacity of 10 items stored at data[0] through data[9]. If the stack currently holds 10 elements, determine the outcome of performing a Push operation to add a new element.

Andaikan tindakan dilaksanakan menggunakan tatasusunan dengan kapasiti 10 item yang disimpan pada data[0] melalui data[9]. Jika tindakan pada masa ini memegang 10 elemen, tentukan hasil melaksanakan operasi Tekan untuk menambah elemen baharu.

- A. The new element will be added at data[10].
Elemen baharu akan ditambah di data[10].
- B. The stack will automatically resize to accommodate the new element.
Stack akan mengubah saiz secara automatik untuk memuatkan elemen baharu.
- C. An overflow condition will occur.
Keadaan overflow akan berlaku.
- D. The stack will overwrite the last element.
Stack akan menulis ganti elemen terakhir.

CLO1

14. Select the situation that represents First In First Out (FIFO) operation in real life.
Pilih situasi yang menerangkan operasi First In First Out (FIFO) dalam situasi kehidupan sebenar.

- A. People waiting in line at a coffee shop.
Orang-orang yang menunggu dalam barisan di kedai kopi.
- B. People standing around a table, taking turns to eat.
Orang-orang berdiri mengelilingi meja, mengambil giliran untuk makan.
- C. People sitting around in a circle having a conversation.
Orang-orang yang duduk dalam bulatan berbual.
- D. People waiting for an elevator to go up.
Orang-orang yang menunggu lif untuk naik.

CLO1

15. Identify the operation used to add an element to the rear of a queue.
Kenalpasti operasi yang digunakan untuk menambah elemen di belakang baris gilir.

- A. Enqueue
- B. Dequeue
- C. Push
- D. Pop

- CLO1 16. The characters of the word **P, O, S, I, T, I, V, E** are placed in a queue in sequence order and then removed one at a time. Trace the order of the characters when they are removed.

*Huruf dalam perkataan **P, O, S, I, T, I, V, E** dimasukkan ke dalam baris gilir mengikut susunan, dan kemudian dikeluarkan satu demi satu. Jejak susunan huruf apabila ia dikeluarkan.*

- A. P, O, S, I, T, I, V, E
- B. P, O, S, I, T, V, I, E
- C. E, V, I, T, I, S, O, P
- D. E, I, V, T, I, S, O, P

- CLO1 17. Figure A17 shows a sequence of operations performed on a queue. Predict the next element that will be removed if one more dequeue() operation is performed.
Rajah A17 menunjukkan satu urutan operasi yang dilakukan ke atas giliran (queue). Ramalkan elemen seterusnya yang akan dikeluarkan jika satu lagi operasi dequeue() dilakukan.

a) Enqueue("F");
b) Enqueue("Y");
c) Enqueue("P");
d) Dequeue();
e) Enqueue("S");

Figure A17 / Rajah A17

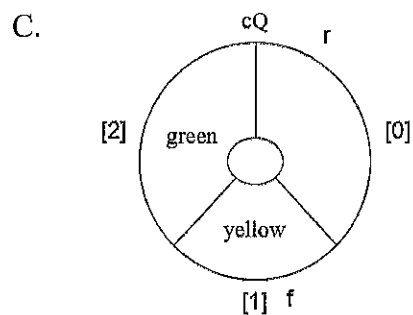
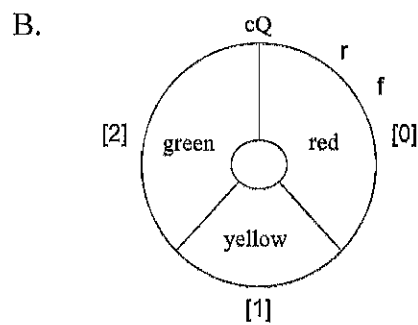
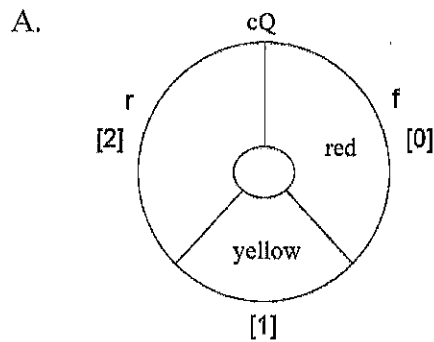
- A. F
- B. Y
- C. P
- D. S

CLO1 18. Choose the **CORRECT** answer of queue after the operations in Figure A18 executed.

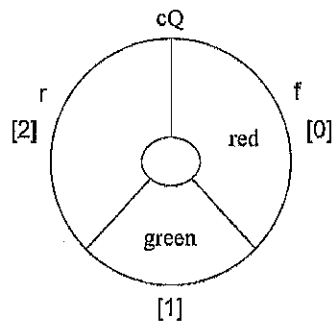
*Pilih jawapan yang **BETUL** untuk baris gilir selepas operasi dalam Rajah A18 dilaksanakan.*

- a) Create Circular Queue(cQ)
- b) Enqueue(cQ, "red");
- c) Enqueue(cQ, "yellow");
- d) Dequeue(cQ);
- e) Enqueue(cQ, "green");

Figure A18 / Rajah A18



D.



CLO1

19. Figure A19 shows a circular queue implemented using an array. Determine the **CORRECT** value for each front, rear and count variables after instruction **enqueue(45)** is executed.

*Rajah A19 menunjukkan satu baris gilir membulat yang dilaksanakan menggunakan tatasusunan. Tentukan nilai **BETUL** bagi setiap pembolehubah front, rear, dan count selepas arahan **enqueue(45)** dilaksanakan.*

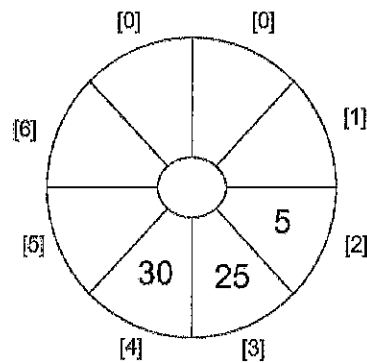


Figure A19 / Rajah A19

- A. front = 1, rear = 5, count = 4
- B. front = 2, rear = 6, count = 5
- C. front = 6, rear = 2, count = 4
- D. front = 2, rear = 6, count = 4

CLO1

20. Name the top node in a tree.
Namakan nod teratas dalam pepohon.

- A. Parent node
- B. Root node
- C. Leaf node
- D. Child node

CLO1

21. Select the **CORRECT** statements when inserting a new node into a Binary Search Tree (BST).
*Pilih pernyataan yang **BETUL** apabila memasukkan nod baharu ke dalam Pepohon Carian dedua (BST).*
- i. The new node is always inserted as a leaf node.
Nod baharu sentiasa disisipkan sebagai nod daun.
 - ii. If the new node's value is smaller than the current node, it is inserted in the left subtree.
Jika nilai nod baharu lebih kecil daripada nod semasa, ia disisipkan dalam subpokok kiri.
 - iii. If the new node's value is greater than the current node, it is inserted in the right subtree.
Jika nilai nod baharu lebih besar daripada nod semasa, ia disisipkan dalam subpokok kanan.
 - iv. A new node can be inserted anywhere in the tree without following an order.
Nod baharu boleh disisipkan di mana-mana dalam pepohon tanpa mengikut sebarang aturan.
- A. i, ii and iii
i, ii dan iii
- B. i, ii and iv
i, ii dan iv
- C. i, iii and iv
i, iii dan iv
- D. ii, iii and iv
ii, iii dan iv

CLO1

22. Indicate the total number of leaf nodes in a full and complete Binary Tree if the height is 3.
Tentukan jumlah nod daun dalam pepohon dedua penuh dan lengkap dengan ketinggian 3.
- A. 4
- B. 6
- C. 8
- D. 10

CLO1

23. Determine node A and node B for Binary Search Tree based on Figure A23.
Tentukan nod A dan nod B untuk Pepohon Carian Dedua berdasarkan Rajah A23.

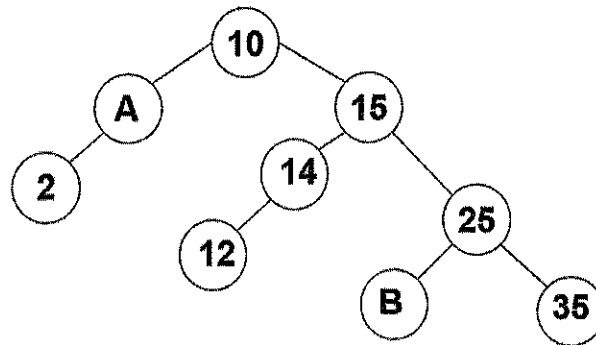


Figure A23 / Rajah A23

- A. A = 13, B = 20
- B. A = 1, B = 21
- C. A = 6, B = 22
- D. A = 8, B = 37

CLO1

24. Choose a suitable node that can replace node Q if node Q is deleted based on Figure A24.
Pilih nod yang sesuai yang boleh menggantikan nod Q jika nod Q dihapuskan berdasarkan Rajah A24.

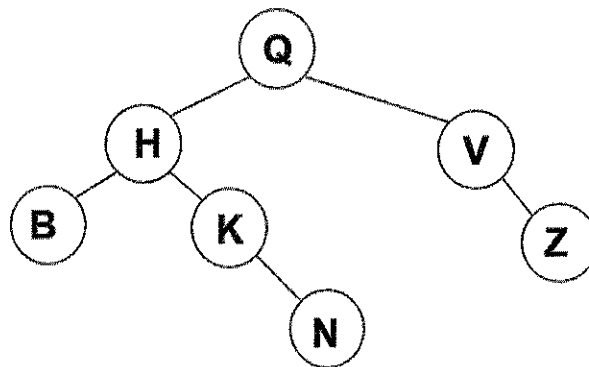


Figure A24 / Rajah A24

- A. N
- B. H
- C. V
- D. B

CLO1

25. Determine the **POSTORDER** traversal of the binary search tree in Figure A25. *Tentukan **POSTORDER** bagi pepohon carian binari dalam Rajah A25.*

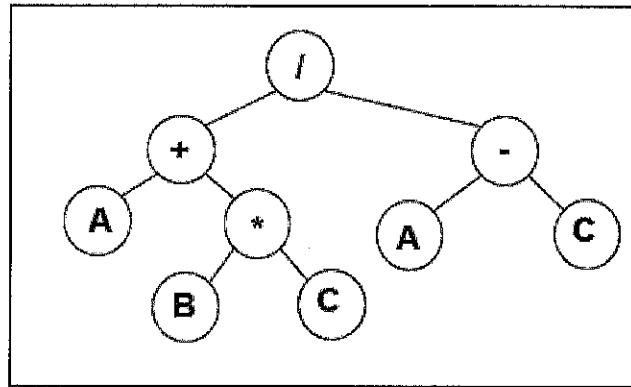


Figure A25 / Rajah A25

- A. $A B C * + A C - /$
- B. $A B + C * A - C /$
- C. $B C * A + A C - /$
- D. $B A * C + A C - /$

CLO1

26. Determine the location of the **new node 28** in Figure A26. Figure A26 shows the original binary search tree. *Tentukan lokasi **nod baharu 28** dalam Rajah A26. Rajah A26 menunjukkan pokok carian binari asal.*

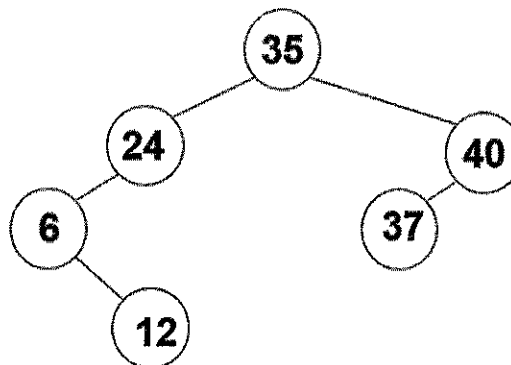
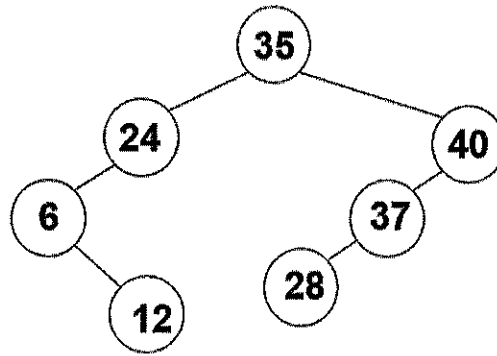
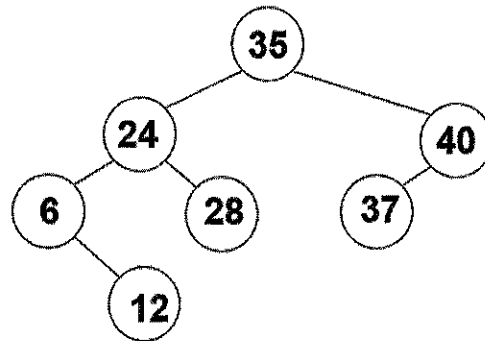


Figure A26 / Rajah A26

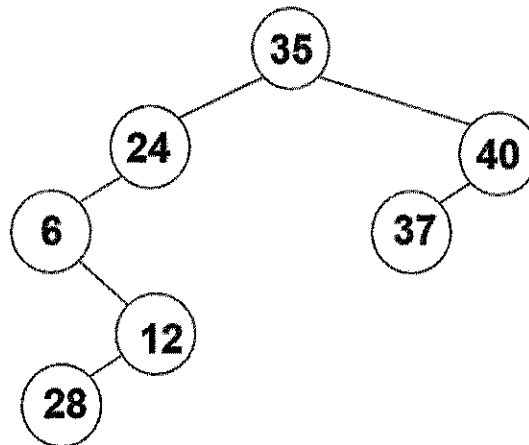
A.



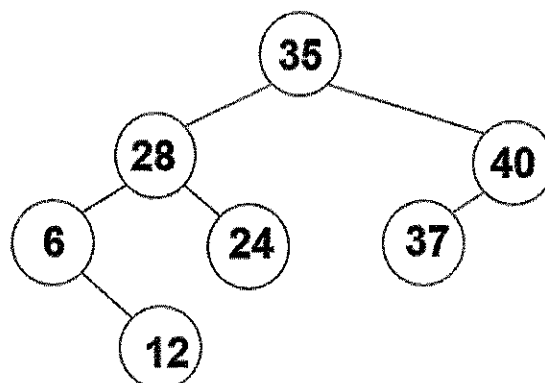
B.



C.



D.



- CLO1 27. Select the **CORRECT** sorting algorithm that applies the divide and conquer method.
Pilih algoritma isihan yang betul yang menggunakan kaedah 'divide and conquer'.

- A. Merge sort.
Isihan gabungan.
- B. Bubble sort
Isihan bebuih.
- C. Insertion sort.
Isihan masukkan.
- D. Selection sort
Isihan pilihan.

- CLO1 28. Based on Figure A28, identify the **middle value** using Binary Search.
Berdasarkan Rajah A28, kenal pasti nilai tengah menggunakan Carian Binari.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
3	9	18	21	27	36	45	58	77

Figure A28 / Rajah A28

- A. 27
- B. 36
- C. 45
- D. 58

- CLO1 29. Based on the unsorted list in Figure A29, identify the number of comparisons using the Linear Search method when searching for the character 'E'.
Berdasarkan senarai tidak tersusun dalam Rajah A29, kenal pasti jumlah perbandingan yang diperlukan menggunakan kaedah Carian Selari apabila mencari karakter 'E'.

Unsorted List: M O N S T E R Y

Figure A29 / Rajah A29

- A. 2
- B. 4
- C. 5
- D. 6

- CLO1 30. Based on the list of numbers below, the first two elements have been sorted using Insertion Sort. Determine the element values in the array after two more sorting steps are implemented.
Berdasarkan senarai nombor di bawah, dua elemen pertama telah disusun menggunakan Insertion Sort. Tentukan nilai elemen dalam tatasusunan selepas dua lagi langkah isihan dilaksanakan.

Data list = { 8, 12, 4, 7, 15, 20, 10 }

- A. { 8, 12, 7, 4, 15, 20, 10 }
- B. { 4, 8, 12, 7, 15, 20, 10 }
- C. { 4, 7, 8, 12, 15, 20, 10 }
- D. { 8, 4, 7, 12, 15, 20, 10 }

SECTION B : 55 MARKS**BAHAGIAN B : 55 MARKAH****INSTRUCTION:**

This section consists of **TWO (2)** structured questions. Answer **ALL** questions.

ARAHAN :

Bahagian ini mengandungi DUA (2) soalan berstruktur. Jawab SEMUA soalan.

QUESTION 1(a)**SOALAN 1(a)**

CLO1

- i. State **TWO (2)** primitive data types that can be used in Linear Data Structures.

Nyatakan DUA (2) jenis data primitif yang boleh digunakan dalam Struktur Data Linear.

[2 Marks]

[2 Markah]

CLO1

- ii. Based on the information in Table B1(a)(ii), show an appropriate **structure declaration** to store 40 student records, with the **structure named StudentRecord** and the **structure variable DIT3A**.

Berdasarkan maklumat dalam Jadual B1(a)(ii), tunjukkan satu perisytiharan struktur yang sesuai untuk menyimpan rekod 40 pelajar, dengan nama struktur StudentRecord dan pembolehubah struktur DIT3A.

Table B1(a)(ii)/ Jadual B1(a)(ii)

Field	Data Type
student_name	50 characters
student_ID	15 characters
CGPA	Floating point

[4 Marks]

[4 Markah]

QUESTION 1(b)**SOALAN 1(b)**

CLO1

- i. List **TWO (2)** operations of Linked List.

Senaraikan DUA (2) operasi Senarai Berpaut.

[2 Marks]

[2 Markah]

CLO1

- ii. A linked list consists of two nodes: B and C, where B is the first node and C is the next node. Illustrate a diagram to show the insertion of a new node A at the beginning of the linked list.

Senarai terpaut terdiri daripada dua nod: B dan C, di mana B adalah nod pertama dan C adalah nod seterusnya. Ilustrasikan satu rajah untuk menunjukkan penyisipan nod baharu A di permulaan senarai terpaut.

[4 Marks]

[4 Markah]

QUESTION 1(c)

SOALAN 1(c)

CLO1

- i. Define the concept of a Stack in Data Structures.

Takrifkan konsep Stack dalam Struktur Data.

[2 Marks]

[2 Markah]

CLO1

- ii. Give **ONE (1)** example of a real-life application and explanation of stack including how the LIFO principle is used.

*Berikan **SATU (1)** contoh aplikasi kehidupan sebenar dan penjelasan tindakan termasuk cara prinsip LIFO digunakan.*

[2 Marks]

[2 Markah]

CLO1

- iii. Given that the stack size is 4, illustrate an appropriate stack diagram for each of the following operations based on Figure B1(c)(iii).

Saiz tindakan yang diberikan ialah 4. Berdasarkan Rajah B1(c)(iii), lukiskan rajah tindakan yang sesuai bagi setiap operasi berikut:

- a) PUSH(18,H);
b) PUSH(4, H);
c) POP(H);
d) PUSH(24,H);

Figure B1(c)(iii) / Rajah B1(c)(iii)

[4 Marks]

[4 Markah]

QUESTION 2(a)**SOALAN 2(a)**

CLO1

- i. Define the given terms below:

Takrifkan istilah yang diberikan dibawah:

- a) Enqueue
- b) Dequeue

[4 Marks]

[4 Markah]

CLO1

- ii. Construct the following operations step by step and illustrate the queue using a linked list representation, including the Front and Rear pointers, based on Figure B2(a)(ii).

Bina operasi berikut secara berperingkat dan ilustrasikan giliran menggunakan perwakilan senarai terpaut, termasuk penunjuk Front dan Rear, berdasarkan Rajah B2(a)(ii).

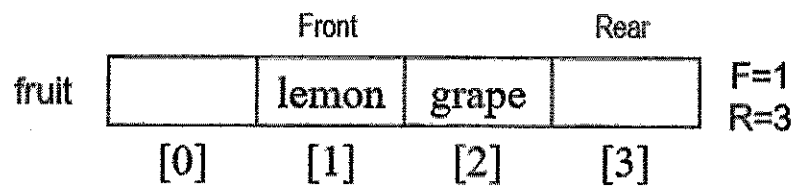


Figure B2(a)(ii) / Rajah B2(a)(ii)

- a) Enqueue("cherry", fruit);
- b) Dequeue(fruit);
- c) Enqueue("kiwi", fruit);

[6 Marks]

[6 Markah]

QUESTION 2(b)**SOALAN 2(b)**

CLO1

- i. Identify parent node, child node, leaf node and depth of tree based on Figure B2(b)(i).

Kenal pasti nod induk (parent node), nod anak (child node), nod daun (leaf node), dan kedalaman pokok (depth of tree) berdasarkan Rajah B2(b)(i).

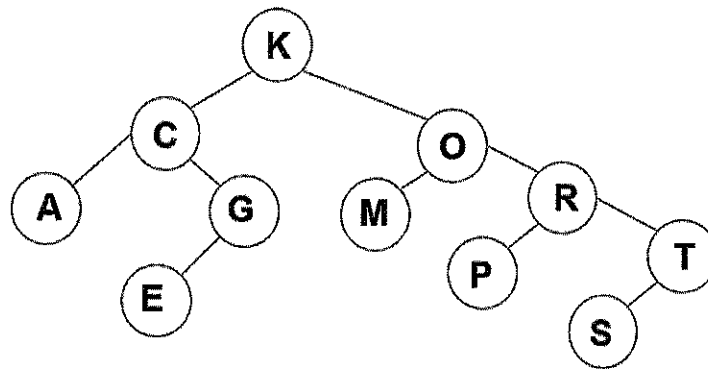


Figure B2(b)(i) / Rajah B2(b)(i)

[4 Marks]

[4 Markah]

CLO1

- ii. Sketch the Binary Tree based on the arithmetic operation in Figure B2(b)(ii).

Lakarkan Pepohon Dedua berdasarkan ungkapan aritmatik dalam Rajah B2(b)(ii).

$$(5 + 2) * (8 - 3) + 6$$

Figure B2(b)(ii) / Rajah B2(b)(ii)

[6 Marks]

[6 Markah]

CLO1

- iii. Write the node sequence of Inorder, Preorder and Postorder traversal based on Binary Tree in Figure B2(b)(iii).

Tulis urutan nod bagi traversal Inorder, Preorder dan Postorder berdasarkan Pepohon Ddua dalam Rajah B2(b)(iii).

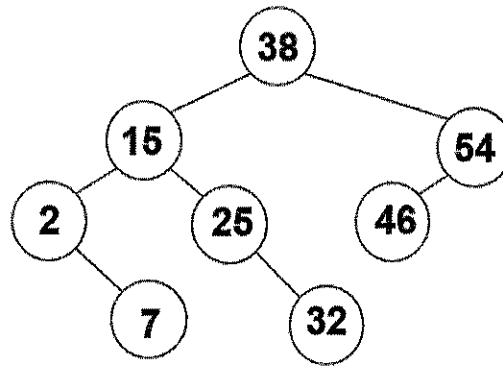


Figure B2(b)(iii) / Rajah B2(b)(iii)

[6 Marks]

[6 Markah]

QUESTION 2(c)**SOALAN 2(c)**

CLO1

- i. Explain **TWO (2)** differences between Linear Search and Binary Search.

*Terangkan **DUA (2)** perbezaan antara Carian Linear dan Carian Binari.*

[4 Marks]

[4 Markah]

CLO1

- ii. Based on the list in Figure B2(c)(ii), apply **Selection Sort** in **ascending order** and show the steps involved.

*Berdasarkan senarai dalam Rajah B2(c)(ii), gunakan **Selection Sort** dalam tertib **menaik** dan tunjukkan langkah-langkah yang terlibat.*

Given List :

27, 76, 14, 25, 21, 38

Figure B2(c)(ii) / *Rajah B2(c)(ii)*

[5 Marks]

[5 Markah]

SOALAN TAMAT