



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Sekolah Pendidikan
Profesional dan
Pendidikan
Berterusan
(SPACE)

**FINAL EXAMINATION / PEPERIKSAAN AKHIR
SEMESTER I – SESSION 2023 / 2024 / SEMESTER I – SESI 2023 / 2024
PROGRAM KERJASAMA**

COURSE CODE : DDWD 2733
KOD KURSUS

COURSE NAME : DATA STRUCTURE AND ALGORITHMS /
NAMA KURSUS STRUKTUR DATA DAN ALGORITMA

YEAR / PROGRAMME : 2 / DDWD
TAHUN / PROGRAM

DURATION : 2 HOURS 30 MINUTES
TEMPOH 2 JAM 30 MINIT

DATE : DECEMBER 2023 / JANUARY 2024
TARIKH DISEMBER 2023 / JANUARI 2024

INSTRUCTION : ANSWER ALL QUESTIONS IN THE QUESTION PAPER.

ARAHAN : JAWAB SEMUA SOALAN DALAM KERTAS SOALAN.

(You are required to write your name and your lecturer's name on your answer script)
(Pelajar dikehendaki tuliskan nama dan nama pensyarah pada skrip jawapan)

NAME / NAMA PELAJAR	:
I.C NO. / NO. K/PENGENALAN	:
YEAR / PROGRAMME TAHUN / PROGRAM	:
COLLEGE NAME NAMA KOLEJ	:
LECTURER'S NAME NAMA PENSYARAH	:

This examination paper consists of 18 pages including the cover
Kertas soalan ini mengandungi 18 muka surat termasuk kulit hadapan



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

School of
Professional and
Continuing
Education
(SPACE)

PUSAT PRGORAM KERJASAMA

PETIKAN DARIPADA PERATURAN AKADEMIK ARAHAN AM – PENYELEWENGAN AKADEMIK

1. SALAH LAKU SEMASA PEPERIKSAAN

1.1. Pelajar tidak boleh melakukan mana-mana salah laku peperiksaan seperti berikut :-

- 1.1.1. memberi dan/atau menerima dan/atau memiliki sebarang maklumat dalam bentuk elektronik, bercetak atau apa jua bentuk lain yang tidak dibenarkan semasa berlangsungnya peperiksaan sama ada di dalam atau di luar Dewan/Bilik Peperiksaan melainkan dengan kebenaran Ketua Pengawas; atau
- 1.1.2. menggunakan maklumat yang diperoleh seperti di atas bagi tujuan menjawab soalan peperiksaan; atau
- 1.1.3. menipu atau cuba untuk menipu atau berkelakuan mengikut cara yang boleh ditafsirkan sebagai menipu semasa berlangsungnya peperiksaan; atau
- 1.1.4. lain-lain salah laku yang ditetapkan oleh Universiti (seperti membuat bising, mengganggu pelajar lain, mengganggu Pengawas menjalankan tugasnya).

2. HUKUMAN SALAH LAKU PEPERIKSAAN

2.1. Sekiranya pelajar didapati telah melakukan pelanggaran mana-mana peraturan peperiksaan ini, setelah diperakukan oleh Jawatankuasa Peperiksaan Fakulti dan disabitkan kesalahannya, Senat boleh mengambil tindakan dari mana-mana satu yang berikut :-

- 2.1.1. memberi markah SIFAR (0) bagi keseluruhan keputusan peperiksaan kursus yang berkenaan (termasuk kerja kursus); atau
 - 2.1.2. memberi markah SIFAR (0) bagi semua kursus yang didaftarkan pada semester tersebut.
- 2.2. Jawatankuasa Akademik Fakulti boleh mencadangkan untuk diambil tindakan tatatertib mengikut peruntukan Akta Universiti dan Kolej Universiti, 1971, Kaedah-kaedah Universiti Teknologi Malaysia (Tatatertib Pelajar-pelajar), 1999 bergantung kepada tahap kesalahan yang dilakukan oleh pelajar.
- 2.3. Pelajar yang didapati melakukan kesalahan kali kedua akan diambil tindakan seperti di perkara dan dicadang untuk diambil tindakan tatatertib mengikut peruntukan Akta Universiti dan Kolej Universiti, 1971, Kaedah-kaedah Universiti Teknologi Malaysia (Tatatertib Pelajar-pelajar), 1999.

SECTION A: TRUE OR FALSE [10 MARKS]

BAHAGIAN A: BENAR ATAU SALAH [10 MARKAH]

1. Random To access the fifth element in a singly linked list which using array implementation, program must first traverse the first four elements.

Untuk mencapai elemen kelima dalam senarai berpaut tunggal yang dilaksanakan menggunakan tatasusunan, aturcara mesti terlebih dahulu melawati empat elemen pertama dahulu.

2. During a searching operation, linear search only needs one loop to complete the task while binary search needs a nested loop to do that process.

Semasa operasi carian, carian linear hanya memerlukan satu gelung untuk menyelesaikan tugas manakala carian binari memerlukan satu gelung bersarang untuk melaksanakan proses tersebut.

3. Total maximum nodes for complete binary with height of 7 is 64.

Jumlah node maksima untuk pepohon binary lengkap dengan ketinggian 7 adalah 64.

4. An binary tree must have at least minimum two sub tree.

Pepohon binari mestilah mempunyai sekurang-kurangnya minimum dua sub pepohon.

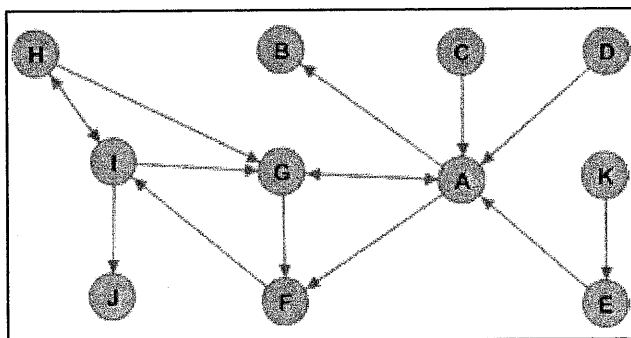
5. In the double linked list, the last node's link points to the first node of the list.

Dalam senarai berpaut berganda, bahagian berpaut nod yang terakhir menunjuk kepada nod yang pertama dalam senarai.

6. In a general list, data can be inserted and deleted anywhere and there are no restrictions on the operations that can be used to process the list.

Dalam senarai berjenis umum, data boleh dimasukkan dan dibuang di mana-mana dan tidak ada sekatan terhadap operasi-operasi yang boleh digunakan untuk memproses senarai.

- 7.



Based on the above graph, the outdegree for G is 3.

Berdasarkan graf di atas, keluaran bagi G ialah 3

8. Data cannot be sorted in descending sequence.

Data tidak boleh disusun di dalam jujukan menurun.

9. If the data in an array is unsorted, binary search algorithm is more efficient than sequential search algorithm.

Jika data di dalam tatasusunan tidak tersusun, algoritma carian binari lebih efektif berbanding dengan algoritma carian jujukan.

10.

```
void recursiveFunction()
{
    recursiveFunction();
}
int main()
{
    recursiveFunction();
    return 0;
}
```

Based on the above fragment code, the function will be called continuously until the stack overflows and there is no more space to store the function calls.

Berdasarkan keratan kod di atas, fungsi akan dipanggil secara berterusan sehingga timbunan melimpah dan tiada lagi ruang untuk menyimpan panggilan fungsi.

SECTION B : MULTIPLE CHOICES [30 MARKS]

BAHAGIAN B : ANEKA PILIHAN [30 MARKAH]

1.

```
void recursiveFunction(int n)
{
    if(n == 0)
        return;
    cout << n << " ";
    recursiveFunction(n-2);
}
int main()
{
    recursiveFunction(20);
    return 0;
}
```

What is the output of the above code?

Apakah hasil keluaran kod di atas?

- A. 0 2 4 6 ... 18 20
- B. 20 18 16 14 ... 2 0
- C. 20 18 16 14 ... 4 2
- D. 2 4 6 8 ... 18 20

2. Which of the following statements about linked list delete is FALSE.

Yang manakah di antara berikut adalah pernyataan PALSU mengenai pembuangan senarai berpaut.

- A. Deletion of a node from the middle of the list requires that its predecessor to be changed.
Membuang satu nod di tengah senarai memerlukan node sebelumnya ditukar.
- B. Deletion of the first node requires that the head pointer to be changed.
Membuang nod pertama memerlukan kepala penunjuk ditukar.
- C. Deletion of the last node requires a separate test to set the predecessor's link node to 0.
Membuang nod terakhir memerlukan pengujian yang berasingan untuk menyetkan pautan node sebelum kepada 0.
- D. The deleted node must be recycled.
Nod yang dibuang mesti diguna semula.

3. Given the following set of data.

Diberi set data berikut.

[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
6	10	26	33	52	78	89	92	98

Using binary search to find target value 89. The search will start at element ___.

Menggunakan carian binari untuk mencari nilai sasaran 89. Carian akan bermula pada elemen ___.

- A. 1
- B. 4
- C. 5
- D. 7

4. Choose correct information about 'Create List' algorithm.

Pilih maklumat yang benar mengenai algoritma 'Membina Senarai'.

- i. At the end of the process, head node is allocated to the list.
Pada akhir proses, nod kepala dimasukkan ke dalam senarai.
- ii. A newly created list is actually an empty list.
Senarai yang baru dibina ialah suatu senarai kosong.
- iii. There is no node in the list, so the count is set to zero.
Tidak terdapat sebarang nod dalam senarai, maka medan 'count' diset kepada sifar.
- iv. The head node pointer is set to null.
Penunjuk nod kepala diset kepada 'tiada nilai'.

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

5. Choose the TRUE statement from the followings:

Pilih pernyataan yang BENAR daripada yang berikut:

- i. A header node in linked list is physically positioned so that it is always the first node in the linked list.
Nod kepala dalam senarai berpaut secara fizikalnya diposisikan supaya ia sentiasa menjadi nod pertama dalam senarai berpaut.
- ii. Stack count function returns the number of elements currently in the stack.
Fungsi 'Stack count' mengembalikan bilangan elemen terkini di dalam timbunan.
- iii. A pseudocode statement $pNew \rightarrow count = 0$ means that the linked-list is empty.
Pernyataan pseudokod $pNew \rightarrow count = 0$ bermaksud senarai berpaut adalah kosong.

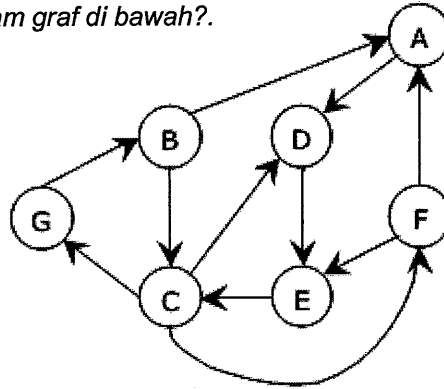
- A. i, ii, iii
- B. i, ii
- C. ii, iii
- D. i, iii

6. Two "start pointer", first node and last node. Forward pointer of the last node points to the first node and backward pointer of the first node points to the last node. Both statements refer to
Dua "penunjuk mula", nod pertama dan nod terakhir. Penunjuk kehadapan nod terakhir menunjuk kepada nod pertama dan penunjuk kebelakang nod pertama menunjuk kepada nod terakhir. Kedua-dua pernyataan tersebut merujuk kepada

- A. Singly linked list / *Senarai berpaut 'singly'*
- B. Circular singly linked list / *Senarai berpaut 'singly' berpusing*
- C. Doubly linked list / *Senarai berpaut 'doubly'*
- D. Circular doubly linked list / *Senarai berpaut 'doubly' berpusing*

7. How cycle inside the graph below?

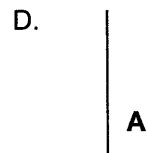
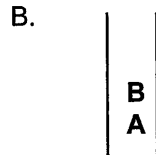
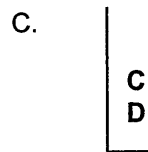
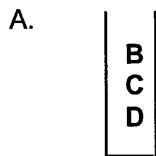
Berapakah kitaran dalam graf di bawah?



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

8. If you **push** the letters A, B, C and D in order into a stack of character and then **pop** two times before calling **stack top** operation. What will the stack look like at the end of the process?

*Jika anda **push** abjad A, B, C dan D mengikut turutan ke dalam satu timbunan aksara dan kemudian **pop** dua kali sebelum memanggil operasi **stack top**. Apakah timbunan yang akan kelihatan di akhir proses?*



9. Which algorithm could effectively solve the problem of 'Towers of Hanoi'?

Algoritma manakah boleh menyelesaikan masalah 'Towers of Hanoi' dengan berkesan?

- A. Doubly-linked list / *Senarai berpaut 'doubly'*
- B. Stack and Queue / *Timbunan dan Baris*
- C. Sorting / *Isihan*
- D. Recursive / *Ulangan*

10. Choose the correct prefix statement for infix statement below.

Pilih pernyataan prefik yang betul untuk pernyataan infik di bawah.

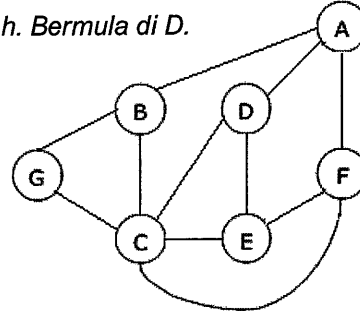
$$P - (R + S * N) / F$$

- A. - / + * P R S N F
- B. - P / + R * S N F
- C. P R S N * + F / -
- D. - / + * R S N P F

11. Give Breadth First Traversal for the graph below. Starting at D.

Berikan Penjelajahan Keluasan Pertama untuk graf di bawah. Bermula di D.

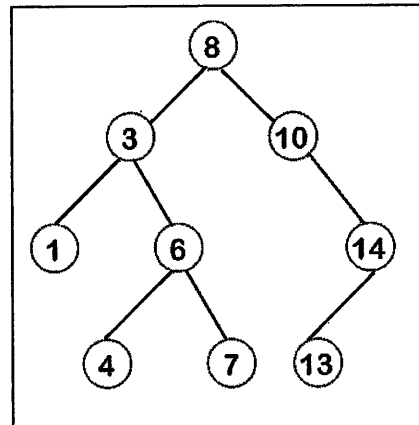
- A. D A C E B F G
- B. D E F C G B A
- C. D E F C A B G
- D. D A C E G B A



12. What is the value of the balance factor for the tree below?

Apakah nilai faktorimbangan bagi pepohon di bawah?

- A. -1
- B. 0
- C. 1
- D. 2



13. Using the following data below, create a max heap tree.

Menggunakan senarai data berikut di bawah, buat pepohon timbunan maks.

102, 204, 308, 405, 106, 207, 57, 60

What kind of tree that will be displayed?

Apakah jenis pepohon yang akan dipamerkan?

- A. Unbalance Binary Tree / Pepohon Binari Tidak Seimbang
- B. Complete Binary Tree / Pepohon Binari Lengkap
- C. Nearly Complete Binary Tree / Pepohon Binari Hampir Lengkap
- D. Binary Tree with height of 3 / Pepohon Binari dengan ketinggian 3

14. Which function will be call to examines the element at the front of the queue?

Fungsi manakah yang akan dipanggil untuk memeriksa elemen pada hadapan baris gilir?

- A. Enqueue
- B. Dequeue
- C. Queue rear
- D. Queue front

15. What is the final time complexity for the code below?

Apakah kerumitan masa akhir untuk kod di bawah?

```
n =1000;  
for ( i=1; i <= n; i++ )  
    for ( j=1; j <= n; j=j*2 )  
        count++;
```

- A. $O(n)$
- B. $O(n^2)$
- C. $O(\log n)$
- D. $O(n \log n)$

ANSWER SECTION A AND B / RUANGAN JAWAPAN BAHAGIAN A DAN B

Answers for Section A [10M] Jawapan untuk Bahagian A:	
Question / Soalan	Answer / Jawapan
Example/Contoh:	TRUE
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Answers for Section B [30M] Jawapan untuk Bahagian B:	
Question / Soalan	Answer / Jawapan
Example/Contoh:	A
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

SECTION C: STRUCTURE [50 MARKS]
BAHAGIAN C: STRUKTUR [50 MARKAH]

QUESTION / SOALAN 1

- a) Based on your understanding, give a simple explanation of what is the data structure. [2M]

Berdasarkan pemahaman anda, berikan penerangan ringkas tentang apakah itu struktur data.

Answer / Jawapan:

- b) What are the differences between data structure and algorithm? [2M]

Apakah perbezaan diantara struktur data dan algoritma?

Answer / Jawapan:

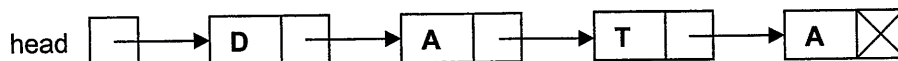
QUESTION / SOALAN 2

- a) What is the output display if the following function is called in **main()** by passing into it the following linked list below with the first node pointed by the head? [2M]

*Apakah paparan output jika fungsi berikut dipanggil dalam **main()** dengan menghantarkan padanya senarai terpaut di bawah dengan nod pertama ditunjuk oleh kepala?*

```
void functionOne(node* head)
{
    if(head == NULL)
        return;

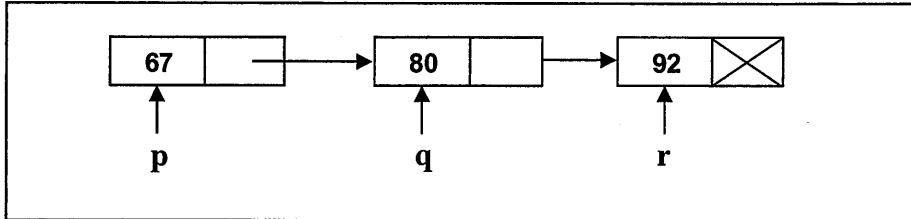
    functionOne(head->next);
    cout << head->data << endl;
}
```



Answer / Jawapan:

b) Imagine that you have 3 variables pointer p, q and r and also 2 nodes with data field and next fields. Based on the following diagram, give result output after statement cout being executed. (Each statement is not related). [3M]

Andaikan anda mempunyai 3 pembolehubah penunding p, q dan r serta 2 nod yang mempunyai medan data dan medan nod berikut. Berdasarkan kepada rajah berikut, berikan hasil keluaran setelah penyataan **cout** dilaksanakan (Setiap penyataan adalah tidak berkaitan antara satu sama lain).



i. `p = q->next; cout << p->data;`

Answer / Jawapan:

ii. `q = p->next; cout << q->data;`

Answer / Jawapan:

iii. `r = p; cout << r->data`

Answer / Jawapan:

QUESTION / SOALAN 3

a) Give content values inside Q1, Q2, S, X, Y and Z below after the command is executed. [6M]

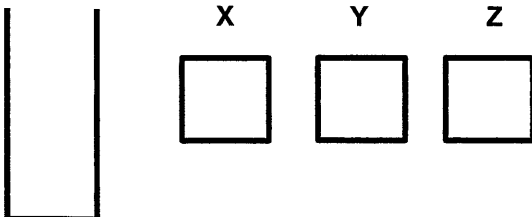
Berikan kandungan nilai dalam Q1, Q2, S, X, Y dan Z selepas arahan tersebut dilaksanakan.

Answer / Jawapan:

Q1 _____

Q2 _____

S



```

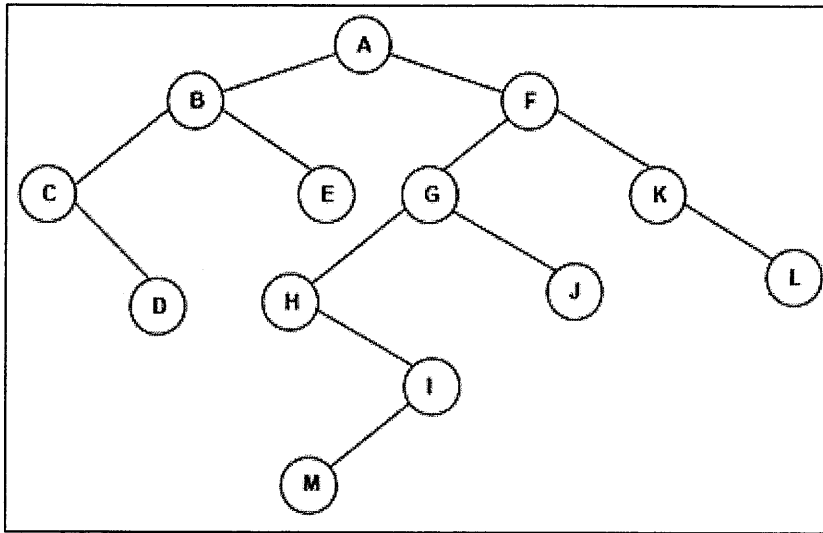
Q1=createQueue
Q2=createQueue
S=createStack
enqueue(Q1, 5)
enqueue(Q1, 3)
enqueue(Q2, 7)
enqueue(Q2, 4)
queueRear(Q1, X)
queueFront(Q2, Y)
pushStack(S, X)
pushStack(S, Y)
loop (not emptyQueue Q2)
    dequeue(Q2, Z)
    Z=Z+X
    pushStack(S, Z)
endLoop
popStack(S, Z)
enqueue(Q2, topStack(S, Y)+2)
  
```


QUESTION / SOALAN 4

a) Given to you the following binary tree. Write down the following traversal.

[4M]

Diberikan kepada anda pepohon binari berikut. Tuliskan rentasan-rentasan berikut.



i. Inorder Traversal

Answer / Jawapan:

ii. Postorder Traversal

Answer / Jawapan:

iii. Preorder Traversal

Answer / Jawapan:

iv. Breadth First Traversal

Answer / Jawapan:

b) Refer to binary tree above (a), answer questions below.

[2M]

Rujuk pepohon di atas (a), jawab soalan-soalan di bawah.

i. leaf nodes / nod-nod daun

Answer / Jawapan:

ii. height of the tree / ketinggian pepohon

Answer / Jawapan:

c) Draw expression tree for expression below.

[3M]

Lukiskan pepohon ungkapan untuk ungkapan di bawah.

$$a + (b - c * (d + e) / f)$$

Answer / Jawapan:

- d) Show the result of inserting below list of sequence number into an initially binary search tree. [3M]
Tunjukkan hasil memasukkan senarai nombor jujukan di bawah ke dalam pepohon carian binari yang pada mulanya kosong.

24, 13, 22, 45, 30, 56, 20, 45, 50

Answer / Jawapan:

- e) Based on above tree (d). What is the value of balance factor for the tree? [1M]
Berdasarkan pepohon di atas (d). Nyatakan nilai faktor keseimbangan untuk pepohon itu.

Answer / Jawapan:

- f) Write algorithm for **Post Order Traversal** for a binary tree. [3M]
*Tulis algoritma untuk **Post Order Traversal** untuk pokok binari.*

Answer / Jawapan:

QUESTION / SOALAN 5

- a) Consider the following algorithm.
What would be returned if **functionTwo** is called in each statement below. [2M]
*Pertimbangkan algoritma berikut.
Apakah yang akan dipulangkan jika **functionTwo** dipanggil dalam setiap pernyataan di bawah.*

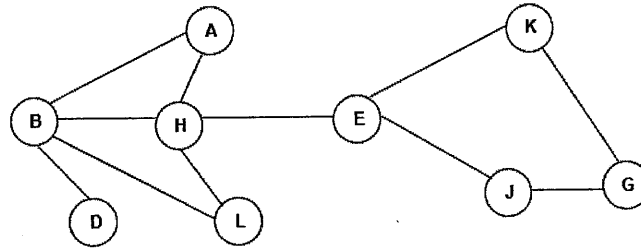
```
Algorithm functionTwo ( x <integer>, y <integer> )
1  If (x < y)
   1  return 2
2  else
   1  return (functionTwo( x-2, y+1 ) + x)
3  end if
```

- i. **functionTwo (8, 3)** **Answer / Jawapan:**
ii. **functionTwo (3, 8)** **Answer / Jawapan:**

b) Give **Breadth First Traversal** for the graph below.

[3M]

*Berikan **Penjelajahan Lebaran Pertama** untuk graf di bawah.*



Answer / Jawapan:

QUESTION / SOALAN 6

a) An array contains the elements shows below. Choose between sequential or binary search algorithm to find 56. Show the tracing steps in each loop iteration. [3M]

Satu array mengandungi elemen-elemen dipaparkan di bawah. Pilih antara algoritma carian jujukan atau binari untuk mencari 56. Tunjukkan jejak langkah yang diikuti dalam setiap ulangan lelaran.

8 13 17 30 37 44 56 63 69 71 88 97

Answer / Jawapan:

- b) Given a list of numbers: 24 18 55 2 94 31 45. Sort the list (by drawing a sort diagram for each phase) by using bubble sort. **[4M]**

Diberi suatu senarai nombor: 24 18 55 2 94 31 45. Isih senarai ini (dengan melukiskan rajah isihan untuk setiap fasa) dengan menggunakan isihan buih.

Answer / Jawapan:

ORIGINAL: 24 18 55 2 94 31 45

SECTION D: PROGRAMMING AND ALGORITHM [10 MARKS]
BAHAGIAN D: PENGATURCARAAN DAN ALGORITMA [10 MARKAH]

Given to you declaration class structure in STACK.h and QUEUE.h below. Write testApplication.cpp to solve question below using combination stack and queue structure. There will one queue object and two stack object.

Read a sentence, enqueue each of the character into a queue. Each time reading the character check it whether it is consonant or vowel. Push consonant character into stack consonant and push vowel character into vowel stack. Inside Queue there were only consonant character leave inside. Example of the output shown as below.

Diberikan kepada anda struktur pengisytiharan kelas dalam STACK.h dan QUEUE.h di bawah. Tulis testApplication.cpp untuk menyelesaikan soalan di bawah menggunakan struktur timbunan dan giliran.

Akan ada satu objek giliran dan dua objek timbunan

Baca satu ayat, masukkan setiap aksara ke dalam giliran. Setiap kali membaca aksara, periksa sama ada ia konsonan atau vokal. Masukkan aksara konsonan ke dalam timbunan konsonan dan masukkan aksara vokal ke dalam timbunan vokal. Di dalam giliran hanya ada watak konsonan yang tersisa di dalamnya.

Contoh output ditunjukkan seperti di bawah.

STACK.h

```
class STACK{  
  
    private:  
        NODE *top;  
        int count;  
  
    public:  
        STACK();  
        bool stackEmpty();  
        bool push(DATA dataIn);  
        bool pop(DATA &dataOut);  
        bool stackTop(DATA &dataOut);  
};
```

QUEUE.h

```
class QUEUE{  
  
    private:  
        NODE *front;  
        NODE *rear;  
        int count;  
  
    public:  
        QUEUE();  
        bool stackEmpty();  
        bool enqueue(DATA dataIn);  
        bool dequeue(DATA &dataOut);  
        bool queueFront(DATA &dataOut);  
        bool queueRear(DATA &dataOut);  
};
```

Insert a sentence: Congratulation You Win

Display Queue: CngrtlttnYWn

Display Vowel Stack: iuooiauaao

Display Consonant Stack: nWYntlttrgnC

Answer / Jawapan: