



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Sekolah Pendidikan Profesional dan
Pendidikan Berterusan
(SPACE)

**FINAL EXAM / PEPERIKSAAN AKHIR
SEMESTER I – SESSION 2021 / 2022
PROGRAM KERJASAMA**

COURSE CODE : DDWD2733 / DDWC2733 / DDPC 2423 / DDC2423
KOD KURSUS

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

YEAR / PROGRAMME : 2 DDWD / DDWC / DDWZ
TAHUN / PROGRAM

DURATION : 3 HOURS (INCLUDING SUBMISSION HOUR)
TEMPOH : 3 JAM (TERMASUK MASA PENGHANTARAN)

DATE : JANUARY 2022
TARIKH

INSTRUCTION / ARAHAN

1. The question paper consists of 4 sections: A, B, C and D.
Kertas soalan terdiri daripada 4 bahagian: A, B, C dan D.
2. Answer **ALL** questions in the answer script attachment.
*Jawab **SEMUA** soalan dalam lampiran skrip jawapan.*
3. Write your name, matric no., identity card no., course code, course name, section no. and lecturer's name on the first page (in the upper left corner) and every page thereafter on the answer sheet.
Tulis nama anda, no. matrik, no. kad pengenalan, kod kursus, nama kursus, no. seksyen dan nama pensyarah pada muka surat pertama (penjuru kiri atas) kertas jawapan dan pada setiap muka surat jawapan.
4. Each answer sheet must have a page number written at the bottom right corner.
Setiap helai kertas jawapan mesti ditulis nombor muka surat pada bahagian bawah penjuru kanan.
5. Answers should be handwriting, neat and clear.
Jawapan hendaklah ditulis tangan, kemas dan jelas menggunakan huruf cerai.

WARNING / AMARAN

Students caught copying / cheating during the examination will be liable for disciplinary actions and the faculty may recommend the student to be expelled from sitting for exam.
Pelajar yang ditangkap meniru / menipu semasa peperiksaan akan dikenakan tindakan disiplin dan pihak fakulti boleh mengesyorkan pelajar diusir dari menduduki peperiksaan.

This examination paper consists of 15 pages including the cover
Kertas soalan ini mengandungi 15 muka surat termasuk kulit hadapan

ONLINE EXAMINATION RULES AND REGULATIONS
PERATURAN PEPERIKSAAN SECARA DALAM TALIAN

1. Student must carefully listen and follow instructions provided by invigilator.
Pelajar mesti mendengar dan mengikuti arahan yang diberikan oleh pengawas peperiksaan dengan teliti.
2. Student is allowed to start examination only after confirmation of invigilator if all needed conditions are implemented.
Pelajar dibenarkan memulakan peperiksaan hanya setelah pengesahan pengawas peperiksaan sekiranya semua syarat yang diperlukan telah dilaksanakan.
3. During all examination session student has to ensure, that he is alone in the room.
Semasa semua sesi peperiksaan pelajar harus memastikan bahawa dia bersendirian di dalam bilik.
4. During all examination session student is not allowed to use any other devices, applications except other sites permitted by course lecturer.
Sepanjang sesi peperiksaan pelajar tidak dibenarkan menggunakan peranti dan aplikasi lain kecuali yang dibenarkan oleh pensyarah kursus.
5. After completing the exam student must inform invigilator via the set communication platform (eg. WhatsApp etc.) about completion of exam and after invigilator's confirmation leave examination session.
Selepas peperiksaan selesai, pelajar mesti memaklumkan kepada pengawas peperiksaan melalui platform komunikasi yang ditetapkan (contoh: Whatsapp dan lain-lain) mengenai peperiksaan yang telah selesai dan meninggalkan sesi peperiksaan selepas mendapat pengesahan daripada pengawas peperiksaan.
6. Any technical issues in submitting answers online have to be informed to respective lecturer within the given 30 minutes. Request for re-examination or appeal will not be entertain if complains are not made by students to their lecturers within the given 30 minutes.
Sebarang masalah teknikal dalam menghantar jawapan secara dalam talian perlu dimaklumkan kepada pensyarah masing-masing dalam masa 30 minit yang diberikan. Permintaan untuk pemeriksaan semula atau rayuan tidak akan dilayan sekiranya aduan tidak dibuat oleh pelajar kepada pensyarah mereka dalam masa 30 minit yang diberikan.
7. During online examination, the integrity and honesty of the student is also tested. At any circumstances student is not allowed to cheat during examination session. If any kind of cheating behaviour is observed, UTM have a right to follow related terms and provisions stated in the respective Academic Regulations and apply needed measures.
Semasa peperiksaan dalam talian, integriti dan kejujuran pelajar juga diuji. Walau apa pun keadaan pelajar tidak dibenarkan menipu semasa sesi peperiksaan. Sekiranya terdapat sebarang salah laku, UTM berhak untuk mengikuti terma yang dinyatakan dalam Peraturan Akademik.

Excerpts from online final exam guidelines

Petikan daripada panduan peperiksaan akhir dalam talian

Universiti Teknologi Malaysia

SECTION A: TRUE FALSE [10 MARKS]

BAHAGIAN A: BENAR SALAH [10 MARKAH]

1. Structure means a set of rules that groups the data together.
Struktur memberi maksud satu set peraturan yang membolehkan data dikumpulkan bersama.
2. Inserting a last **node** in the linked list share the same logic as inserting a **node** at the beginning of linked list.
*Kemasukan satu **nod** di akhir senarai berpaut berkongsi logik yang sama seperti memasukkan satu **nod** di permulaan senarai berpaut.*
3. By using an Abstract Data Type (**ADT**), programmers can hide implementation details of program codes from the user.
*Dengan menggunakan Jenis Data Abstrak (**ADT**), pengaturcara boleh menyembunyikan implementasi kod-kod aturcara daripada pengguna.*
4. Four basic queue operations are **Enqueue, Queue Top, Queue Front, Queue Rear**.
*Empat operasi asas giliran ialah **Enqueue, Queue Top, Queue Front, Queue Rear**.*
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. The **outdegree** of a vertex in a **digraph** is the number of arcs entering the vertex.
***Luadarjah** bagi satu nod di dalam satu graf berarah adalah bilangan arka yang memasuki nod.*
8. If the data in an array is sorted, binary search algorithm is more efficient than sequential search algorithm.
Jika data di dalam tatasusunan telah tersusun, algoritma carian binari lebih efektif berbanding dengan algoritma carian jujukan.
9. Data cannot be sorted in descending sequence.
Data tidak boleh disusun di dalam jujukan menurun.
10. Traversal is a repetitive process in which an algorithm calls itself.
Penjelajahan adalah proses yang berulang di mana suatu algoritma memanggil dirinya sendiri.

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

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

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

SECTION B : MULTIPLE CHOICES [30 MARKS]

BAHAGIAN B : ANEKA PILIHAN [30 MARKAH]

1. Which of the following is **NOT** a step in adding a **node** to a linked list?
*Manakah di antara berikut **BUKAN** satu langkah di dalam memasukkan satu **nod** ke satu senarai berpaut?*
 - A. Determine the insertion point / *Memastikan titik masukkan*
 - B. Point the new node to the successor / *Nod baru menunjukkan kepada nod selepas*
 - C. Point the predecessor to the new node / *Nod sebelum menunjukkan kepada nod baru*
 - D. Traverse the list / *Senarai dijelajah*

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 to 0.
Membuang nod terakhir memerlukan pengujian yang berasingan untuk menyetkan 'link' node sebelum kepada 0.
 - D. The deleted node must be recycled.
Nod yang dibuang mesti diguna semula.

3. A linear list where all addition and deletion are restricted to only one end refers to
Satu senarai berpaut di mana kesemua kemasukan dan pembuangan adalah terhad hanya pada satu hujung merujuk kepada
 - A. linked list / *Senarai berpaut*
 - B. Stack / *Timbunan*
 - C. Queue / *Giliran*
 - D. Tree / *Pepohon*

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->count = 0` means that the linked-list is empty.
Penyataan pseudokod `pNew->count = 0` bermaksud senarai berpaut adalah kosong.

- A. i, ii
- B. i, iii
- C. ii, iii
- D. i, ii, 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 hadapan 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. Choose right answer for postfix statement below.

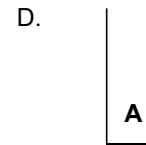
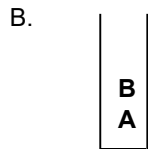
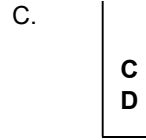
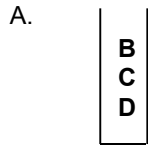
Pilih jawapan yang betul untuk ungkapan posfik dibawah.

$$3 \ 2 \ 4 \ + \ 10 \ - \ *$$

- A. 36
- B. 30
- C. 16
- D. -12

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 efektif?

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

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

Pilih pernyataan postfix yang betul untuk pernyataan infik di bawah.

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

- A. $PR - SN * F / +$
- B. $PRSN * F / - +$
- C. $PRSN * + F / -$
- D. $PRSN + F * / -$

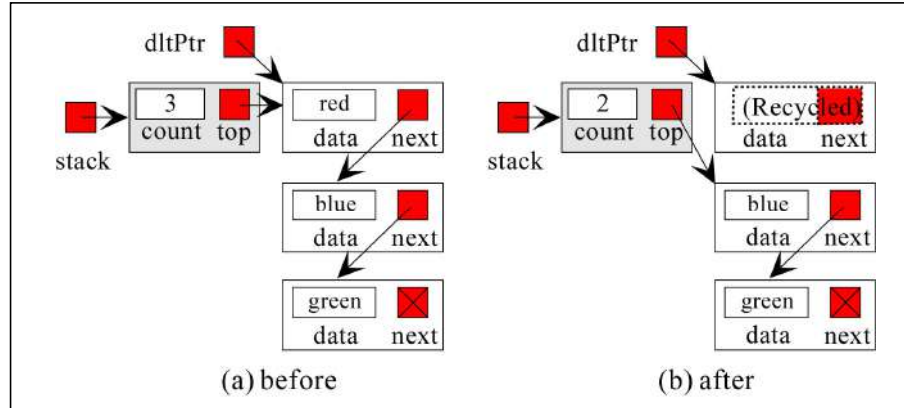
11. Which operation can insert an element after the head node of the data structure?

Operasi manakah boleh menambah elemen selepas nod kepala struktur data?

- A. Linear list and Stack / Senarai linear dan Timbunan
- B. Stack and Queue / Timbunan dan Baris
- C. Linear list and Queue / Senarai Linear dan Baris
- D. Stack and Tree / Timbunan dan Pokok

12. Given a diagram of a logical operation in processing stack structure. Study the diagram carefully and choose which option defines true about the diagram.

Diberi rajah yang menunjukkan operasi logikal dalam memproses struktur data timbunan. Analisa rajah dan pilih yang manakah benar menerangkan rajah ini.



- A. Empty stack
B. Stack top
C. Push stack
D. Pop stack
13. Which definition defines best about the **level** of a node in a tree structure?
Definisi manakah yang tepat menerangkan mengenai paras suatu nod dalam struktur pepohon?
- A. The level of the node is the distance of the leaf in the longest path from the root.
Paras suatu nod adalah jarak antara daun pada laluan terpanjang dari akar.
- B. The level of the node is the distance of the leaf from the root.
Paras suatu nod adalah jarak antara daun dan akar.
- C. The level of the node is the distance of the leaf from the root plus one.
Paras suatu nod adalah jarak antara daun dan akar ditambah satu.
- D. The level of the node is a sequence of nodes in which each node is adjacent to the next one.
Paras suatu nod adalah turutan nod-nod di mana setiap nod adalah bersebelahan dengan nod selepasnya.
14. _____ function examines the element at the front of the queue.
Fungsi _____ memeriksa elemen pada hadapan baris gilir.
- A. Enqueue
B. Dequeue
C. Queue rear
D. Queue front

15. Listing below shown type linear data structure EXCEPT _____ .
Senarai di bawah menunjukkan jenis struktur data linear KECUALI _____ .
- A. Linear Linked List / *Senarai Berpaut Linear*
 - B. Graph / *Graf*
 - C. Queue / *Baris Gilir*
 - D. Stack / *Timbunan*

SECTION C: STRUCTURE [48 MARKS]

BAHAGIAN C: STRUKTUR [48 MARKAH]

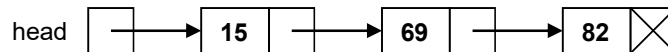
QUESTION / SOALAN 1

- a) Give ONE (1) reason why data structure is important to a programmer? [1M]
Berikan SATU (1) sebab mengapa struktur data adalah penting kepada seorang pengaturcara?
- b) What are the differences between data structure and algorithm? [2M]
Apakah perbezaan diantara struktur data dan algoritma?
- c) Give ONE (1) example how data structure being apply in real life problem. [2M]
Beri SATU (1) contoh bagaimana struktur data diaplikasikan dalam masalah kehidupan sebenar.

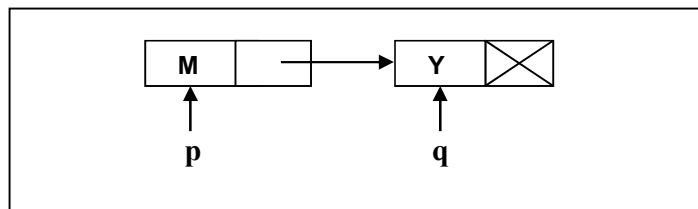
QUESTION / SOALAN 2

- a) What does the following function do for a given following linked list with first node as head? [2M]
Apa yang dibuat oleh fungsi berikut untuk satu senarai berpaut berikut yang diberi dengan node pertama sebagai kepala?

```
void fun1 (node* head)
{
    if (head == NULL)
        return;
    fun1 (head->next);
    cout << head->data << endl;
}
```



- b) Imagine that you have 2 variables pointer, p and q and also 2 nodes with data field and next fields. Based on the following diagram, give result output after statement **cout** being execute. (Each statement is not related). [3 M]
*Andaikan anda mempunyai 2 pembolehubah penunding, p dan q serta 2 nod yang mempunyai medan data dan medan nod berikut. Berdasarkan kepada rajah berikut, berikan hasil keluaran setelah pernyataan **cout** dilaksanakan (Setiap pernyataan adalah tidak berkaitan antara satu sama lain).*



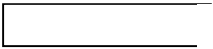
- i. `p = p -> next; cout << p->data;`
- ii. `q -> next = p; cout << q->data;`
- iii. `q = p; cout << q->data`

QUESTION / SOALAN 3

- a) What is the content of Q1, Q2 and S after the command is executed. **[3M]**
Apakah kandungan Q1, Q2 dan S selepas arahan tersebut dilaksanakan.

```
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, X)
enqueue(Q2, topStack(S, Y))
```

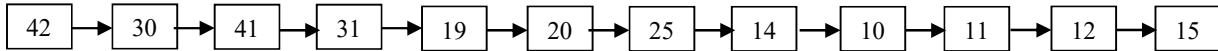
- b) Change the following infix expression to postfix expression using the algorithmic method (a stack). **[5M]**
Tukarkan ungkapan infix berikut kepada ungkapan postfix dengan menggunakan kaedah algoritma (satu tindanan).

infix expression <i>ungkapan infix</i>	Stack <i>Tindanan</i>	postfix expression <i>ungkapan postfix</i>
m / (n + (x - s + y) * z) - t		EMPTY

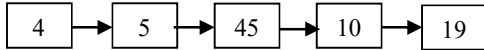
c) Contents of queue Q1 and queue Q2 are as shown below. What would be the content of queue Q3 after the following code is executed? The queue contents are shown front (left) to rear (right). **[2M]**

Kandungan giliran Q1 dan giliran Q2 ditunjukkan seperti di bawah. Apakah kandungan giliran Q3 selepas code berikut dilaksanakan? Kandungan giliran menunjukkan depan (kiri) ke belakang (kanan).

Q1 :



Q2 :



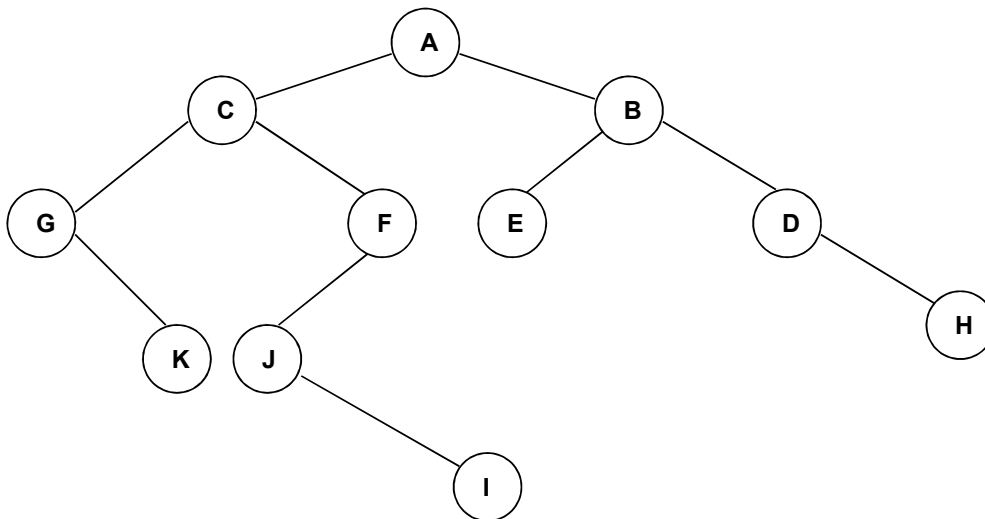
Q3 :

```
1  Q3 = createQueue
2  count = 0
3  loop (not empty Q1 and not empty Q2)
    1  count = count + 1
    2  dequeue (Q1, x)
    3  dequeue (Q2, y)
    4  if (y < x)
        1  enqueue (Q3, x+count)
    5  end if
4  end loop
```

QUESTION / SOALAN 4

a) Given to you the following binary tree. Write down the following traversal. **[8M]**

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



- i. Inorder Traversal**
- ii. Postorder Traversal**
- iii. Preorder Traversal**
- iv. Breadth First Traversal**

- b) Refer to binary tree above (a), identify the nodes below. [2M]

Rujuk pepohon di atas (a), kenalpasti nod-nod di bawah.

- i. internal nodes / *nod-nod dalaman*
- ii. leaf nodes / *nod-nod daun*

- c) Draw expression tree for expression below. [2M]

Lukiskan pepohon ungkapan untuk ungkapan di bawah.

$$A/B-C+D*E/F$$

- d) Show the result of inserting 30, 11, 45, 33, 50, 11, 68 into an initially empty binary search tree. This tree will facilitate duplicate elimination. [2M]

Tunjukkan hasil kemasukan 30, 11, 45, 33, 50, 11, 68 ke dalam satu pepohon binari yang dinilai awalkan kosong. Pepohon ini akan memudahkan penghapusan pendua.

- e) Based on above binary search tree (d). Show the balance factor in the resulting tree. [1M]

Berdasarkan pada pohon carian binari di atas (d). Tunjukkan faktor keseimbangan dalam pepohon yang dihasilkan.

QUESTION / SOALAN 5

- a) Consider the following algorithm.

What would be returned if recFun is called in each statement below. [2M]

Pertimbangkan algoritma berikut.

Apakah yang akan dipulangkan jika recFun dipanggil dalam setiap pernyataan di bawah.

```
Algorithm recFun ( x <integer>, y <integer>)
```

```
1  If (x < y)
```

```
    1  return -3
```

```
2  else
```

```
    1  return (recFun( x-y, y+3 )+y)
```

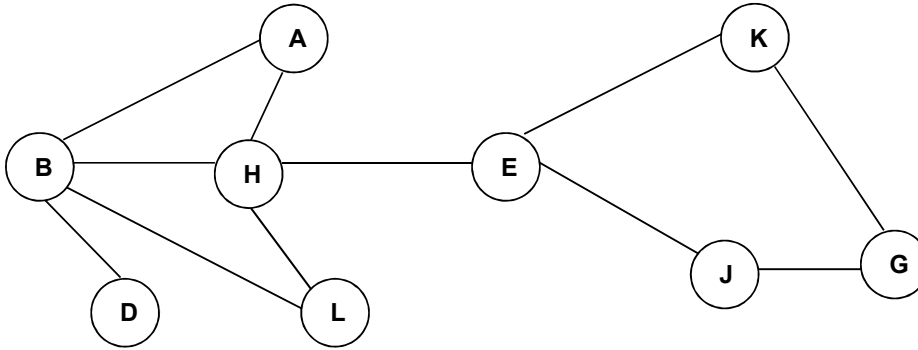
```
3  end if
```

- i. recFun(2, 7)

- ii. recFun(15, 3)

- b) Give **depth first traversal** for the graph below. Please start from H. [3M]

*Berikan **penjelajahan dalaman pertama** untuk graf di bawah. Sila mula dari H.*



QUESTION / SOALAN 6

- a) An array contains the elements shows below. Using the binary search algorithm, trace the steps followed to find 56. At each loop iteration, including the last, show the content of **first**, **last** and **mid**. [2M]

*Satu array mengandungi elemen-elemen dipaparkan di bawah. Gunakan algoritma carian binary, jejak langkah yang diikuti untuk mencari 56. Pada setiap ulangan lelaran, termasuk yang terakhir, tunjukkan kandungan **first**, **last** dan **mid**.*

8 13 17 30 44 56 88 97

- b) Given a list of numbers: 24 55 2 94 31 45. Sort the list (by drawing a sort diagram for each phase) by using selection sort. [3M]

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

ORIGINAL: 24 55 2 94 31 45

- c) Using the same list as in question 5 (b), sort the list by using bubble sort. [3M]

Dengan menggunakan senarai yang sama seperti dalam soalan 5 (b), isih senarai berkenaan dengan menggunakan isihan buih.

ORIGINAL: 24 55 2 94 31 45

SECTION D: PROGRAMMING AND ALGORITHM [12 MARKS]

BAHAGIAN D: PENGATURCARAAN DAN ALGORITMA [12 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	QUEUE.h
<pre>class STACK{ private: NODE *top; int count; public: STACK(); bool stackEmpty(); bool push(DATA dataIn); bool pop(DATA &dataOut); bool stackTop(DATA &dataOut); };</pre>	<pre>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); };</pre>

Insert a sentence: Congratulation You Win

Display Queue: CngrtlttnYWn

Display Vowel Stack: iuooiauo

Display Consonant Stack: nWYntltrgnC