



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Sekolah Pendidikan Profesional dan
Pendidikan Berterusan
(SPACE)

**FINAL EXAM / PEPERIKSAAN AKHIR
SEMESTER II – SESSION 2021 / 2022**

COURSE CODE : 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 DDWC / DDWZ
TAHUN / PROGRAM

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

DATE : JUN / JUL 2022
TARIKH

INSTRUCTION / ARAHAN

1. The question paper consists of 2 sections: A and B.
Kertas soalan terdiri daripada 2 bahagian: A dan B.
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 9 pages including the cover
Kertas soalan ini mengandungi 9 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: STRUCTURE [85 MARKS]

BAHAGIAN A: STRUKTUR [85 MARKAH]

QUESTION / SOALAN 1

a) Why data structure is important to a programmer? [2M]

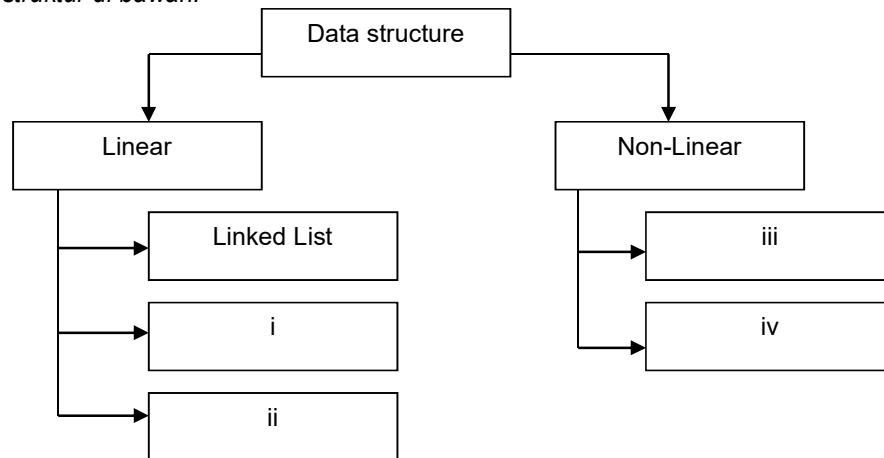
Mengapa struktur data adalah penting kepada seorang pengaturcara?

b) What are the differences between atomic data and composite data? [2M]

Apakah perbezaan diantara data atomic dan data komposit?

c) Please complete the structure below. [2M]

Sila lengkapkan struktur di bawah.



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

Apakah perbezaan diantara struktur data dan algoritma?

e) Give example how data structure being apply in real life problem. [2M]

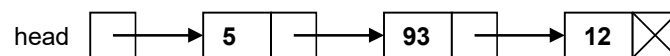
Beri contoh bagaimana struktur data diaplikasikan dalam masalah kehidupan sebenar.

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;
}
```

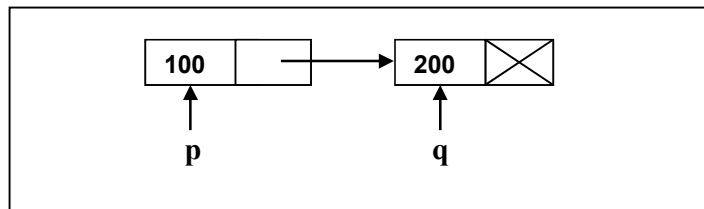


- b) A node in a linked list is a structure that has at least two fields: one contains the data, the other the address of the next node in the sequence. Write down the node declaration in C++. **[3M]**

Satu nod di dalam senarai berpaut adalah satu struktur yang mempunyai paling kurang dua bahagian: satu mengandungi data, dan satu lagi alamat kepada nod yang berikutnya di dalam turutan. Tuliskan pengisytiharan nod di dalam C++.

- c) Imagine that you have 2 variables, p and q and also 2 nodes with data field and next fields. Based on the following diagram, draw an appropriate diagram to illustrate the statements below. (Each statement is not related). **[5 M]**

Andaikan anda mempunyai 2 pembolehubah, p dan q serta 2 nod yang mempunyai medan data dan medan nod berikut. Berdasarkan kepada rajah berikut, lukiskan rajah yang sesuai untuk menggambarkan pernyataan di bawah. (Setiap pernyataan adalah tidak berkaitan antara satu sama lain).



- i. $p = p \rightarrow \text{next}$
- ii. $q \rightarrow \text{next} = p$
- iii. $p \rightarrow \text{next} = q \rightarrow \text{next}$
- iv. $p \rightarrow \text{data} = 50$
- v. $q = p$

QUESTION / SOALAN 3

- a) Draw all the integer Stacks after running the following statement. **[3M]**

Lukiskan kesemua timbunan integer selepas melarikan pernyataan berikut.

```
S1=createStack
S2=createStack
S3=createStack
pushStack(S1,53)
pushStack(S1,14)
pushStack(S1,80)
stackTop(S1,x)
pushStack(S2,x)
pushStack(S2,10)
pushStack(S2,4)
popStack(S1,y)
popStack(S1,x)
pushStack(S3,y)
pushStack(S3,x)
```

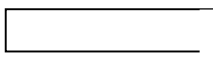
- b) What is the content of Q1, Q2 and Q3 turns after the command is executed. **[6M]**
Apakah kandungan giliran Q1, Q2 dan Q3 selepas arahan tersebut dilaksanakan.

```
Q1=createQueue
Q2=createQueue
Q3=createQueue
enqueue(Q1,5)
enqueue(Q1,3)
enqueue(Q1,6)
enqueue(Q1,8)
enqueue(Q1,0)
enqueue(Q1,12)
enqueue(Q1,15)
enqueue(Q1,0)
enqueue(Q1,5)
loop (not emptyQueue Q1)
    dequeue(Q1,x)
    if (x=0)
        loop (not emptyQueue Q2)
            dequeue(Q2,y)
            z=0;
            z=z+y
            enqueue(Q3,z)
        else
            enqueue(Q2,x)
    endLoop
```

- c) S is a stack of integers and Q is a queue of integers. Draw S and Q after the following operations. **[2M]**
S adalah satu tindanan integer dan Q adalah satu giliran integer. Lukiskan S dan Q selepas operasi berikut.

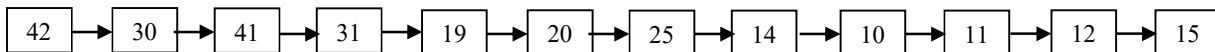
```
1  pushStack (S, 3)
2  pushStack (S, 3)
3  enqueue (Q, 5)
4  enqueue (Q, 8)
5  popStack (S, x)
6  pushStack (S, 2)
7  enqueue (Q, x)
8  dequeue (Q, y)
9  pushStack (S, x)
10 pushStack (S, y)
```

- d) Change the following infix expression to postfix expression using the algorithmic method (a stack). **[4M]**
Tukarkan ungkapan infix berikut kepada ungkapan postfix dengan menggunakan kaedah algoritma (satu tindanan).

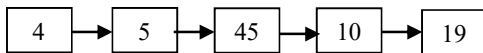
infix expression <i>ungkapan infix</i>	Stack <i>Tindanan</i>	postfix expression <i>ungkapan postfix</i>
m / n + (x - y * z) - t		EMPTY

- e) 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). **[5M]**
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 :



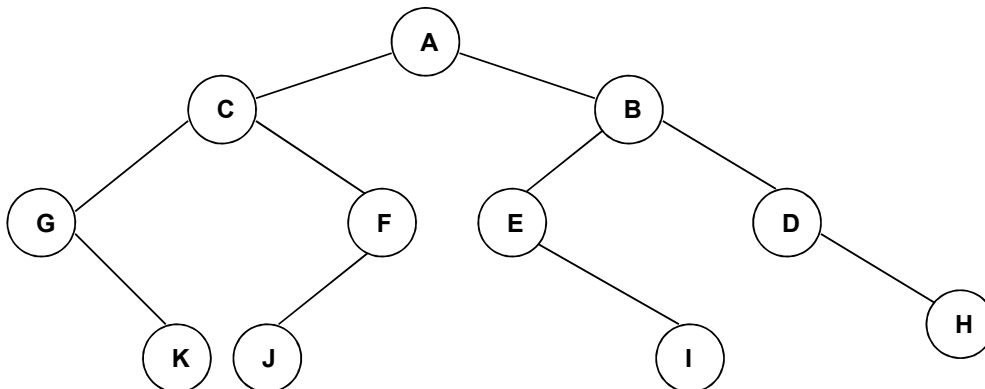
Q3 :

```

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. [8M]

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

- i. internal nodes / *nod-nod dalaman*
- ii. leaf nodes / *nod-nod daun*
- iii. height of subtree B / *ketinggian sub pepohon B*
- iv. level of node F / *aras bagi nod F*

c) Draw expression tree for expression below. [3M]

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. [3M]

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

e) Based on above binary search tree (d). Show the balance factor in the resulting tree. [2M]

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. [3M]

Pertimbangkan algoritma berikut.

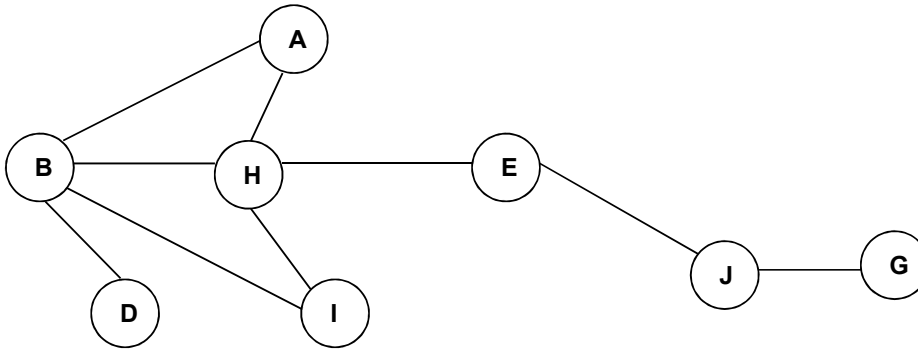
Apakah yang akan dipulangkan jika recFun dipanggil dalam setiap pernyataan di bawah.

```
Algorithm recFun ( x <integer>, y <integer>
5   If (x < y)
6     return -3
6   else
7     return (recFun( x-y, y+3 )+y)
7   end if
```

- i. recFun(2, 7)
- ii. recFun(5, 3)
- iii. recFun(15, 3)

- b) Give **depth first traversal** for the graph below. Please start from H. [4M]

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 88. At each loop iteration, including the last, show the content of **first**, **last** and **mid**. [6M]

*Satu array mengandungi elemen-elemen dipaparkan di bawah. Gunakan algoritma carian binary, jejak langkah yang diikuti untuk mencari 88. 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 -94 45 2 31 55. Sort the list (by drawing a sort diagram for each phase) by using insertion sort. [4M]

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

Answers / Jawapan:

ORIGINAL: 24 -94 45 2 31 55

- c) Using the same list as in question 5 (b), sort the list by using bubble sort. [4M]

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

Answers / Jawapan:

ORIGINAL: 24 -94 45 2 31 55

SECTION B: PROGRAMMING AND ALGORITHM [15 MARKS]

BAHAGIAN B: PENGATURCARAAN DAN ALGORITMA [15 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