



**FINAL EXAM / PEPERIKSAAN AKHIR
SEMESTER I – SESSION 2018 / 2019
PROGRAM KERJASAMA**

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 : 2 HOURS AND 30 MINUTES
TEMPOH

DATE : NOVEMBER 2018
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 question paper.
*Jawab **SEMUA** soalan dalam kertas soalan.*
3. Candidates are required to follow all instructions given by the examination invigilators.
Calon dikehendaki mematuhi semua arahan daripada penyelia peperiksaan.

(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 / COURSE TAHUN / KURSUS	:
COLLEGE KOLEJ	:
LECTURER'S NAME NAMA PENSYARAH	:



PUSAT PROGRAM 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 Peperiksaan melainkan dengan kebenaran Ketua Pengawas; atau
- 1.1.2 menggunakan makluman yang diperolehi 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 2.1.2 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 FALSE [10 MARKS]

BAHAGIAN A : BENAR SALAH [10 MARKAH]

INSTRUCTION: Answer True / False in the provided space at page 7.

ARAHAN: Jawab Benar / Salah di dalam ruangan yang disediakan di muka surat 7.

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. **Overflow** state happen when **pop** operation being done while the stack is empty.
*Status **overflow** terjadi apabila operasi **pop** dilakukan ketika timbunan adalah kosong.*
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. The height of the tree is the level of the leaf in the longest path from the root plus 1
Ketinggian suatu pokok adalah paras daun dalam laluan yang terpanjang daripada akar ditambah 1.
6. An binary tree must have at least minimum two sub tree.
Pepohon binary mestilah mempunyai sekurang-kurangnya minimum dua sub pepohon.
7. The **outdegree** of a vertex in a **digraph** is the number of arcs entering the vertex.
***Luardarjah** 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. Recursive is a repetitive process in which an algorithm calls itself.
Rekursif adalah proses yang berulang di mana suatu algoritma memanggil dirinya sendiri.

SECTION B : MULTIPLE CHOICES [20 MARKS]

BAHAGIAN B : ANEKA PILIHAN [20 MARKAH]

INSTRUCTION: Answer True / False in the provided space at page 7.

ARAHAN: Jawab Benar / Salah di dalam ruangan yang disediakan di muka surat 7.

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 mensetkan '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 refers 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. Choose right answer for postfix statement below.

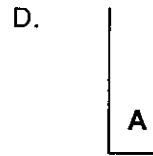
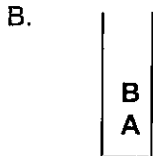
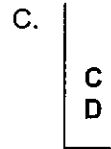
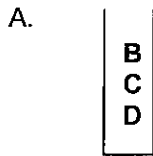
Pilih jawapan yang betul untuk ungkapan posfik dibawah.

6 3 2 4 + - *

- A. 1
- B. 18
- C. -18
- D. 20

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. PRSN - * F / -
- B. PRSN - * F - /
- C. PRSN * - F / -
- D. PRSN - F * / -

ANSWER SECTION / RUANGAN JAWAPAN

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

Answers for Section B [20M] <i>Jawapan untuk Bahagian B:</i>	
Question / Soalan	Answer / Jawapan
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

SECTION C: STRUCTURE [55 MARKS]

BAHAGIAN C: STRUKTUR [55 MARKAH]

Instruction: Write your answers in the spaces provided in this questions paper.

Arahan: Tuliskan jawapan di ruang yang disediakan di dalam kertas soalan ini.

QUESTION / SOALAN 1

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

Mengapa struktur data adalah penting kepada seorang pengaturcara?

Answers / Jawapan:

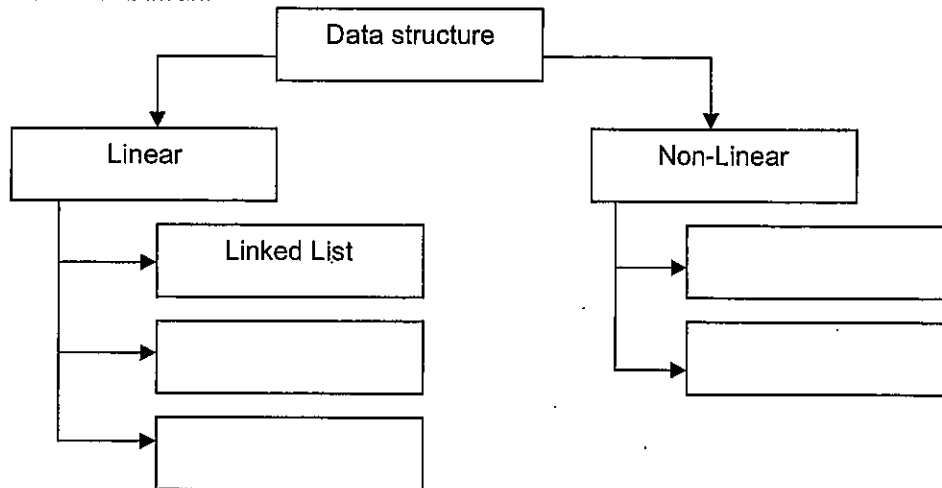
- b) What are the differences between atomic data and composite data? [2M]

Apakah perbezaan diantara data atomic dan data komposit?

Answers / Jawapan:

- 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?

Answers / Jawapan:

QUESTION / SOALAN 2

a) Draw a figures to represent doubly list with each of it containing the integer values 3, 25, 47 and 96. Do not use any dummy nodes. [2M]

Lukis satu rajah untuk mewakili senarai berpaut berganda dengan setiapnya mengandungi nilai integer 3, 25, 47 dan 96. Jangan guna sebarang nod dummy.

Answers / Jawapan:

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++.

Answers / Jawapan:



What happens if you use the following statement in a search of above linked list . [2M]

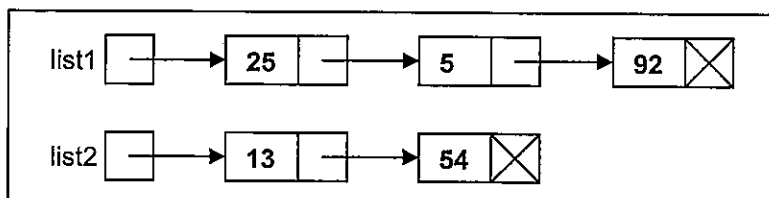
Apakah yang akan berlaku jika anda menggunakan pernyataan berikut di dalam carian senarai berpaut di atas.

```
pLoc = pHead -> link
```

Answers / Jawapan:

d) Imagine you have the two linked lists shown below. What would happen if you applied the following statements? Draw your answer. [3M]

Andaikan anda mempunyai dua senarai berpaut seperti yang ditunjukkan di bawah. Apakah yang akan berlaku sekiranya pernyataan berikut dilaksanakan? Lukiskan jawapan anda.



```
1 temp = list2
2 loop (temp->link not null)
  1 temp = temp->link
3 end loop
4 temp->link = list1
```

Answers / Jawapan:

QUESTION / SOALAN 3

a) Display the output and draw stack S1 after running the following segment of code, given that item1, item2 and item3 are integer variables. [5M]

Paparkan hasil keluaran dan lukis timbunan S1 selepas melarikan segmen kod berikut, diberikan item1, item2 dan item3 adalah pembolehubah integer.

```
Stack S1;
Item1 = 1;
Item2 = 0;
Item3 = 4;
S1.push(item2);
S1.push (item1);
S1.push (item1 + item3);
item2 = S1.pop();
S1.push(item3 * item3);
S1.push(item2);
S1.push(3);
item1 = S1.pop();
cout << item1 << endl << item2 << endl
      << item3 << endl;
```

Answers / Jawapan:

Output:

Stack S1:

- b) 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)

Answers / Jawapan:

infix expression <i>ungkapan infix</i>	Stack <i>Tindanan</i>	postfix expression <i>ungkapan postfix</i>
(A+B)*C-D*F+C	EMPTY	EMPTY

- c) S is a stack of integers and Q is a queue of integers. Draw a diagram of S and Q after the following operations. [3M]

S adalah satu tindanan integer dan Q adalah satu giliran integer. Lukiskan satu rajah 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)

Answers / Jawapan:

QUESTION / SOALAN 4

a) Consider the following algorithm. What would be returned by recFun after statement below. **[2M]**

Pertimbangkan algoritma berikut. Apakah yang akan dipulangkan recFun selepas 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
```

recFun(15, 3)

Answers / Jawapan:

b) Show the result of inserting 13, 11, 41, 60, 9, 12, 25, 17 into an initially empty binary search tree. **[3M]**

Tunjukkan hasil kemasukan 13, 11, 41, 60, 9, 12, 25, 17 ke dalam satu pepohon binari yang dinilaiawalkan kosong.

Answers / Jawapan:

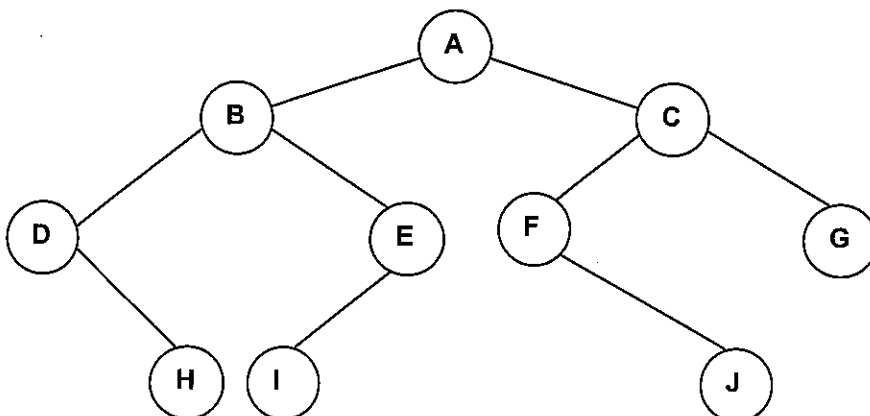
c) Given to you the following binary tree.

Write down the preorder inorder traversal and postorder traversal of that tree.

[4M]

Diberikan kepada anda pepohon binari berikut.

Tuliskan rentasan rentasan ikut-tertib dan rentasan pasca-tertib pepohon tersebut.



Answers / Jawapan:

d) Draw expression tree for expression below.

[3M]

Lukiskan pepohon ungkapan untuk ungkapan di bawah.

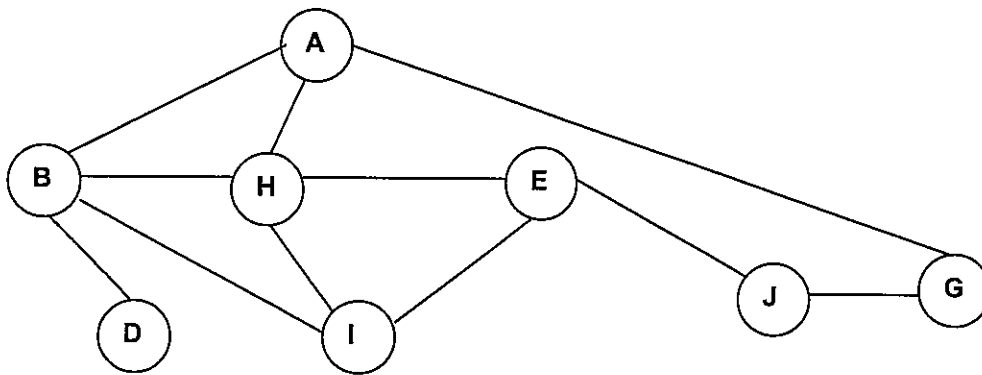
$$A/B-C+D^*E/F$$

Answers / Jawapan:

e) Give depth first traversal for the graph below. Start at node H.

[4M]

Berikan penjelajahan dalaman pertama untuk graf di bawah. Mula pada nod H.



Answers / Jawapan:

QUESTION / SOALAN 5

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**. [3M]

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**.

Answers / Jawapan:

- b) Given a list of numbers: 12 32 8 100 90 45. Sort the list (by drawing a sort diagram for each phase) by using insertion sort. **[3M]**

Diberi suatu senarai nombor: 12 32 8 100 90 45. Isih senarai ini (dengan melukiskan rajah isihan untuk setiap fasa) dengan menggunakan isihan masukKan.

Answers / Jawapan:

- 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.

Answers / Jawapan:

SECTION D: PROGRAMMING [15 MARKS]

BAHAGIAN D: PENGATURCARAAN [15 MARKAH]

Instruction: Write your answers in the spaces provided in this questions paper.

Arahan: Tuliskan jawapan di ruang yang disediakan di dalam kertas soalan ini.

1. Given below stack header file declaration. Using stack, write a C++ function to reverses a list of integers read from the keyboard by pushing them into a stack and retrieving them one by one.

[8M]

Diberi pengisytiharan kepala fail tindanan di bawah. Gunakan tindanan, tulis satu fungsi C++ untuk menterbalikkan satu susunan integer yang di baca daripada papan kekunci dengan memasukkannya ke dalam satu tindanan dan mencapainya satu per satu.

```
class STACK{  
  
    private:  
        node *top;    Node  
        int count;  
  
    public:  
        STACK();  
        bool stackEmpty();  
        void pushStack(DATA dataIn);  
        void popStack(DATA *dataOut);  
        void stackTop(DATA *dataOut);  
};
```

Answers / Jawapan:



2. Base on recursive towers of Hanoi algorithm below and example output solution for three disks.

Give output solution for four disks.

[7M]

Berdasarkan pada algoritma rekusif menara Hanoi di bawah dan contoh paparan penyelesaian untuk tiga cakera.

Berikan paparan penyelesaian menara untuk empat cakera.

```

Algorithm Towers (val disks    <integer>,
                    val source  <character>,
                    val dest    <character>,
                    val auxiliary <character>,
                    ref step    <integer>)
Recursively move disks from source to destination.
Pre   the tower consists of integer disk
        source, destination and auxiliary tower given
Post  steps for moves printed

1   print ("Tower: ", disks , source, dest, auxiliary)
2   if (disks > 1)
    1   print ("Step " , step ,
              "Move from " , source, " to " , dest)
    2   step = step +1
3   Else
    1   towers (disks - 1, source, auxiliary, dest, step)
    2   print ("Step " , step ,
              "Move from " , source, " to " , dest)
    3   step = step +1
    4   towers (disks - 1, auxiliary, dest, source, step)
4   End if
5   return

end towers

```

EXAMPLE OUTPUT / CONTOH PAPARAN

Calls:	Output:
Towers (3, A, C, B)	
Towers (2, A, B, C)	
Towers (1, A, C, B)	
	Step 1: Move from A to C
	Step 2: Move from A to B
Towers (1, C, B, A)	
	Step 3: Move from C to B
	Step 4: Move from A to C
Towers (2, B, C, A)	
Towers (1, B, A, C)	
	Step 5: Move from B to A
	Step 6: Move from B to C
Towers (1, A, C, B)	
	Step 7: Move from A to C

Answers / Jawapan:

- END OF QUESTIONS/ SOALAN TAMAT -

Mukasurat ini sengaja dibiarkan kosong

[This page is purposely left blank]