



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Sekolah Pendidikan Profesional dan
Pendidikan Berterusan
(UTMSPACE)

**FINAL EXAM / PEPERIKSAAN AKHIR
SEMESTER I – SESSION 2017 / 2018
PROGRAM KERJASAMA**

COURSE CODE : DDWC2733 / DDPC 2423 / DDC2423
KOD KURSUS

COURSE NAME : DATA STRUCTURES AND ALGORITHMS / DATA STRUCTURE /
STRUKTUR DATA DAN ALGORITHMMA / STRUKTUR DATA

YEAR / PROGRAMME : 2 DDWC / 2 DDPC / 2 DDPZ
TAHUN / PROGRAM

DURATION : 2 HOURS AND 30 MINUTES
TEMPOH

DATE : OCTOBER / NOVEMBER 2017
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 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	:

This examination paper consists of 12 pages including the cover
Kertas soalan ini mengandungi 12 muka surat termasuk kulit hadapan



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 maklumat 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.

QUESTION / SOALAN 2

a) Show the output of the following segment of code, given that item1, item2 and item3 are integer variables. **[5M]**

Paparkan hasil keluaran 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 << " " << item2 << " " << item3 << endl;
while (!S1.IsEmpty()){
    item1=S1.pop();
    cout << item1 << endl;
}
```

Answers / Jawapan:

b) Give value of the following postfix expressions. Show the steps on how to get those values. **[4M]**

Beri nilai bagi ungkapan posfiks berikut. Tunjukkan langkah-langkah bagaimana mendapatkan nilai tersebut.

Answers / Jawapan:

i. ABC*D+ where / di mana:
A=9, B=2, C=3, D=4

ii. AB-C* where / di mana:
A=3, B=2, C=4

- c) Change the following infix expression to postfix expression using the algorithmic method (a stack). **[7M]**
Tukarkan ungkapan infix berikut kepada ungkapan postfix dengan menggunakan kaedah algoritma (satu tindanan)

Answers / Jawapan:

infix expression / ungkapan infix	stack / tindanan	postfix expression / ungkapan postfix
(A+B)*C-D*F+G	EMPTY	EMPTY

- d) S is a stack of integers and Q is a queue of integers. Draw picture of S and Q after the following operations. **[4M]**

S adalah satu tindanan integer dan Q adalah satu giliran integer. Lukiskan gambaran S dan Q selepas operasi berikut.

- | | |
|----|-------------------|
| 1 | pushStack (S, 3) |
| 2 | pushStack (S, 10) |
| 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) |

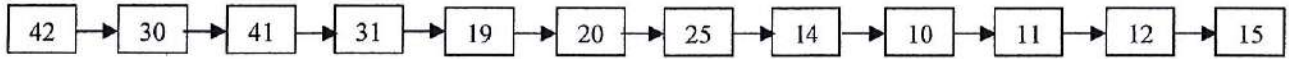
Answers / Jawapan:

e) Contents of queue Q1 and queue Q2 are as shown below. What would be the content of queue Q1, Q2 and 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 Q1, Q2 dan Q3 selepas code berikut dilaksanakan? Kandungan giliran menunjukkan depan (kiri) ke belakang (kanan).

Q1 :



Q2 :



```

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 equal count)
      1  enqueue (Q3, x)
   5  end if
4  end loop
  
```

Answers / Jawapan:

QUESTION / SOALAN 3

a) Consider the following algorithm.

What would be returned if recFun is called in each statement below.

[4M]

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
  
```

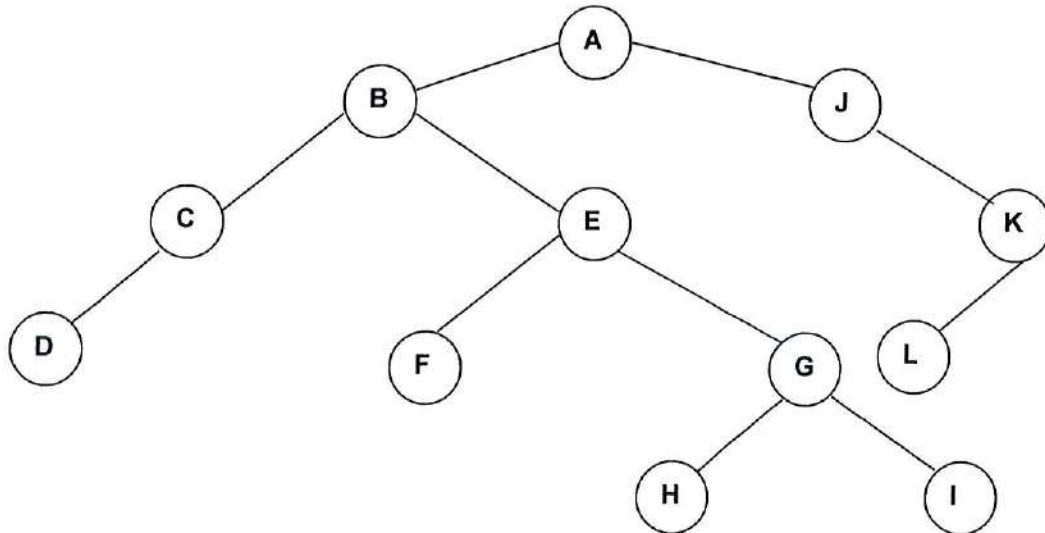
i. recFun(2, 7)

ii. recFun(15, 3)

Answers / Jawapan:

Answers / Jawapan:

QUESTION / SOALAN 4



a) Answer question 1 to 10 based on the binary tree above.

[10M]

Jawab soalan 1 sehingga 10 berdasarkan pada pepohon binari di atas.

No.	Question / Soalan	Answers / Jawapan:
1.	Root of the tree / Akar pepohon	
2.	Leaves / Dedaun	
3.	Internal nodes / nod internal	
4.	Height of the tree / Ketinggian pepohon	
5.	Height of sub tree E / Ketinggian subpepohon E	
6.	Child of B / Anak B	
7.	Level of node G / Aras nod G	
8.	Inorder traversal of tree / Inorder traversal pepohon	
9.	Preorder traversal of tree / Preorder traversal pepohon	
10.	Postorder traversal of tree / Postorder traversal pepohon	

b) Show the result of inserting 36, 17, 13, 49, 24, 52, 40, 77 into an initially empty binary search tree. **[4M]**

Tunjukkan hasil kemasukan 36, 17, 13, 49, 24, 52, 40, 77 ke dalam satu pepohon binari yang dinilaiawalkan kosong.

Answers / Jawapan:

- c) Based on question b above. State down whether the tree is complete binary tree or nearly complete binary tree. [1M]

Berdasarkan soalan b di atas. Nyatakan samada pepohon adalah pepohon binari lengkap atau pepohon binari hampir lengkap.

Answers / Jawapan:

- d) Write binary search tree **inorder** traversal algorithm. [3M]

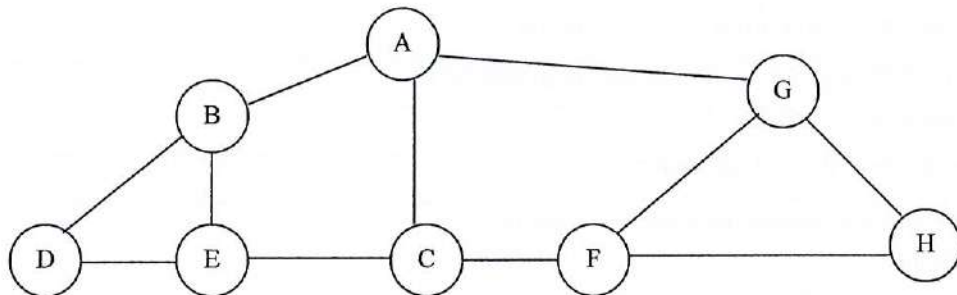
*Tuliskan algoritma pepohon carian binari penjelajahan **inorder**.*

Answers / Jawapan:

QUESTION / SOALAN 5

Answer question a) and b) based on the graph below.

Jawap soalan a) dan b) berdasarkan graf di bawah.



- a) Give the breadth first traversal for the graph above. Show the content of stack or queue. [2 M]

Berikan penjelajahan lebar pertama untuk graf di atas. Tunjukkan isi kandungan tindanan atau giliran.

Answers / Jawapan:

- b) Give the depth first traversal for the graph above. Show the content of stack or queue. [2 M]

Berikan penjelajahan dalaman pertama untuk graf di atas. Tunjukkan isi kandungan tindanan atau giliran.

Answers / Jawapan:

QUESTION / SOALAN 6

a) Name two types of exchange sort.

[2M]

Namakan dua jenis isihan pertukaran.

Answers / Jawapan:

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.

[5M]

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

Answers / Jawapan:

c) Using the same list as in question 5 (b), sort the list by using selection sort.

[5M]

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

Answers / Jawapan:

d) After two phases of a sorting algorithm, the following list 45 9 78 33 55 99 has been rearranged as 9 45 78 33 55 99

Which sorting algorithm is being used (selection, bubble, insertion)?

[2M]

Selepas dua fasa satu algoritma isihan, senarai berikut 45 9 78 33 55 99 telah di susun atur sebagai 9 45 78 33 55 99

Algoritma isihan yang manakah telah digunakan (pilihan, buih, masukkan)?

Answers / Jawapan:

SECTION B: DESIGN [25 MARKS]

BAHAGIAN B: REKABENTUK [25 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++ complete program to reverses a list of string read from the keyboard.

[10M]

Diberi pengisytiharan kepala fail tindanan di bawah. Gunakan tindanan, tulis satu program lengkap C++ untuk menterbalikkan satu susunan string yang di baca daripada papan kekunci.

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

OUTPUT

Insert one word: **HELLO**
Reverse word: **OLLEH**

Answers / Jawapan:

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

i. Give output for 4 disks. [8M]

ii. Draw all the step taken to solve towers of Hanoi for four disks. [7M]

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

i. *Berikan output untuk 4 cakera.*

ii. *Lukiskan kesemua langkah yang perlu diambil untuk menyelesaikan menara Hanoi 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 for 3 disks/
Contoh paparan untuk 3 disk

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:

Mukasurat ini sengaja dibiarkan kosong

[This page is purposely left blank]

Mukasurat ini sengaja dibiarkan kosong

[This page is purposely left blank]