

**SULIT**



**UNIVERSITI TUN HUSSEIN ONN MALAYSIA**

**PEPERIKSAAN AKHIR  
SEMESTER 1  
SESI 2017/2018**

NAMA KURSUS : STRUKTUR DATA DAN  
ALGORITMA  
KOD KURSUS : BBP 25203  
KOD PROGRAM : BBF  
TARIKH PEPERIKSAAN : DISEMBER 2017 / JANUARI 2018  
JANGKA MASA : 3 JAM  
ARAHAN : JAWAB SEMUA SOALAN

**TERBUKA**

KERTAS SOALAN INI MENGANDUNGI **LAPAN (8)** MUKA SURAT

**SULIT**

## BAHAGIAN A

- S1 Manakah antara yang berikut menerangkan satu maklumat tentang individu, tempat atau benda?
- A field
  - B file
  - C record
  - D Tiada dalam jawapan di atas
- S2 Manakah antara berikut merupakan pernyataan benar mengenai tatasusunan (array)?
- A Operasi aritmetik boleh dilakukan dalam tatasusunan
  - B Subskrip terakhir dalam tatasusunan ialah nombor kurang daripada jumlah bilangan lokasi ingatan pembolehkan tatasusunan
  - C Koleksi senarai data di bawah satu nama, dari jenis data yang sama
  - D Setiap elemen dalam tatasusunan diwakili oleh 2 subskrip
- S3 Manakah antara berikut merupakan *scope resolution operator*?
- A :: (two colons)
  - B \* (asterisk)
  - C . (period)
  - D -> (hyphen and greater than symbol)
- S4 Manakah antara berikut memaparkan kandungan pembolehkan `city` ke dalam fail output yang bernama `address.txt`? fail tersebut adalah berkait (associated) dengan objek `outFile`.
- A `address.txt <<city<<endl;`
  - B `ofstream<<city<<endl;`
  - C `outFile<<city<<endl;`
  - D `outFile>>city>>endl;`
- S5 Operasi terhadap data dalam \_\_\_\_\_ secara tatasusunan berlaku pada dua hujung
- A timbunan (stack)
  - B senarai berpaut (linked list)
  - C baris gilir (queue)
  - D pepohon (tree)
- S6 Kemasukan elemen ke dalam tindanan (stack) mesti dibuat dari
- A bahagian atas tindanan (on the top of the stack)
  - B bahagian tengah tindanan (in the middle of the stack)
  - C bahagian bawah tindanan (on the bottom of the stack)
  - D bahagian ruang yang kosong pada tindanan (in any empty spot of a stack)



TERBUKA

- S7 Manakah antara berikut tidak dibenarkan dalam C++?
- A `int &x;`
  - B `char huruf;`
  - C `double _harga;`
  - D `string nama3;`
- S8 Pernyataan "`cin >>dataPelajar.markahUjian`" akan
- A membaca nilai *default* yang ditentukan oleh sistem dan disimpan dalam `markahUjian`
  - B membaca nilai yang dimasukkan oleh pengguna dan disimpan dalam `markahUjian`
  - C membaca nilai yang dimasukkan oleh pengguna dan disimpan dalam `dataPelajar.markahUjian`
  - D membaca nilai *default* yang ditentukan oleh sistem dan disimpan dalam `dataPelajar.markahUjian`
- S9 Algoritma carian binari tidak boleh digunakan untuk
- A senarai berpaut terisih (sorted linked list)
  - B pepohon binari terisih (sorted binary trees)
  - C tatasusunan linear terisih (sorted linear array)
  - D tatasusunan penuding (pointer array)
- S10 Unsur-unsur tatasusunan disimpan secara berturut-turut dalam sel memori kerana
- A dengan cara ini komputer boleh mengesan hanya alamat elemen pertama dan alamat unsur-unsur lain boleh dikira selepas itu
  - B seni bina memori komputer tidak membenarkan tatasusunan untuk menyimpan selain dari secara bersiri
  - C A dan B
  - D Tiada dalam jawapan di atas
- S11 Apakah nilai persamaan postfix  $8\ 2+3\ *16\ 4\ /- ?$
- A -1
  - B 1.625
  - C 10
  - D 26
- S12 Pointer atau penuding
- A merupakan jenis pembolehubah yang berjenis `int`, `float` dan `double`
  - B merupakan jenis data yang menyimpan alamat sebagai nilainya
  - C tidak boleh digunakan bersama tatasusunan
  - D boleh digunakan untuk merujuk tatasusunan secara statik

**TERBUKA**

- S13** Apakah ungkapan infix bagi  $A * B / (C + D) / E$  ?
- A  $A B C D * E / + /$
  - B  $A B * C D + / E /$
  - C  $A B * C D + // E$
  - D  $A * B C D + / E /$
- S14** Manakah antara yang berikut membandingkan nilai ahli dalam pembolehubah `struct pekerja` dan `struct pekerjaSenior`?
- A `if (pekerja.ic == pekerjaSenior.ic && pekerja.nama == pekerjaSenior.nama)`
  - B `if (pekerjaSenior.ic == pekerja.ic && pekerjaSenior.nama == pelajarKanan.nama)`
  - C `if (pekerjaSenior.ic == pekerjaSenior.ic && pekerja.nama == pekerja.nama)`
  - D `if (pekerja.ic == pekerja.nama && pekerjaSenior.ic == pekerjaSenior.nama)`
- S15** Berikut adalah operasi asas yang dilaksanakan ke atas data jenis niskala (abstract data types (ADT)) kecuali
- A konstruktor (constructors)
  - B senarai berpaut (linked list)
  - C pengulangan (iterators)
  - D pemerhatian (observers)
- S16** Pengguna melaksanakan operasi ke atas tindanan bersaiz 5 seperti pernyataan dalam Rajah S16. Berapakah jumlah elemen dalam tindanan tersebut setelah operasi selesai dilaksanakan?

```
push (1)
pop ()
push (2)
push (3)
pop ()
push (4)
pop ()
pop ()
push (5)
```

Rajah S16

- A 1
- B 4
- C 2
- D 3

**TERBUKA**

- S17** Keadaan senarai berpaut apabila `Nod_kepala=NULL` ialah
- A kosong
  - B melimpah
  - C mempunyai satu medan data
  - D tepu
- S18** Kes paling teruk berlaku di dalam satu algoritma carian linear apabila
- A item berada disatu lokasi ditengah-tengah tatasusunan
  - B item tiada di dalam tatatususunan
  - C item ialah elemen terakhir tatasusunan
  - D item ialah elemen terakhir dalam tatatususunan atau tiada dalam tatasusunan
- S19** Semua pernyataan di bawah adalah benar mengenai pepohon kecuali
- A bilangan anak bagi pepohon penduaan (binary tree) boleh melebihi dua
  - B digunakan untuk membuat carian ke atas bilangan data yang besar
  - C pepohon penduaan terdiri dari satu set nod-nod yang terhingga
  - D pepohon bukan kosong mestilah mempunyai nod akar
- S20** Setiap pengisytiharan tatasusunan tidak perlu memberi, secara tersirat atau tersurat, maklumat mengenai
- A nama tatasusunan
  - B jenis data tatasusunan
  - C data yang pertama dari set untuk disimpan
  - D set indeks tatasusunan



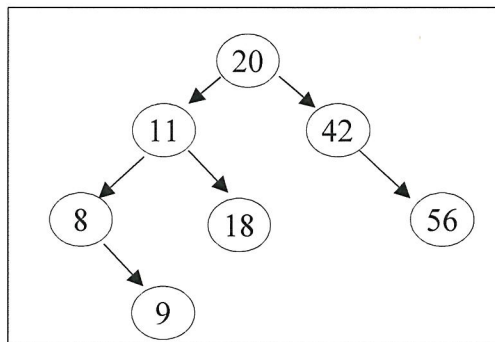
**TERBUKA**

## BAHAGIAN B

- S21**
- a) Berikan satu (1) contoh struktur data linear tidak terhad (linear data structure with no restriction) dan bukan linear (non-linear data structure) (2 markah)
- b) Terangkan kelebihan senarai berpaut yang dilaksanakan secara tatasusunan dan penuding (4 markah)
- c) Terangkan dua (2) operasi asas timbunan (stack) (4 markah)
- S22**
- a) Diberi ungkapan postfix  $7\ 3\ +\ 2\ * =$ , ilustrasikan langkah-langkah penyelesaian ungkapan tersebut dengan menggunakan tindanan (stack) (8 markah)
- b) Tukarkan ungkapan infix berikut kepada ungkapan postfix
- i.  $A * B / (C + D) / E$
- ii.  $A - (B + C) * D + E / F$
- iii.  $A / B + C * D - E$
- (6 markah)
- c) Selesaikan persamaan postfix yang berikut
- i.  $8\ 2\ +\ 3\ * \ 16\ 4\ / \ - =$
- ii.  $3\ 5\ 6\ * \ + \ 13\ - \ 18\ 2\ / \ + =$
- iii.  $12\ 25\ 5\ 1\ // \ * \ 8\ 7\ + \ - =$
- (6 markah)

A red rectangular stamp with the word "TERBUKA" in bold, uppercase letters.

- S23**
- a) Terangkan maksud menyusur pepohon penduaan (traversal in binary trees) dan tujuan melakukan penyusuran (traversal)? (4 markah)
- b) Berdasarkan pepohon pada Rajah S23, senaraikan susunan nod seperti berikut
- Penyusuran sisipan (in-order)
  - Penyusuran awalan (pre-order)
  - Penyusuran akhiran (post-order)



Rajah S23

(6 markah)

- S24** Rajah S24 menunjukkan satu jujukan data yang disimpan di dalam sebuah tatasusunan. Ilustrasikan dengan lukisan langkah-langkah mengisih data tersebut dalam susunan menaik menggunakan *heap sort*

93	82	47	31	64	23	17	27
----	----	----	----	----	----	----	----

Rajah S24

(10 markah)

**TERBUKA**

- S25** Rajah **S25** menunjukkan satu jujukan data yang disimpan di dalam sebuah tatasusunan. Ilustrasikan dengan lukisan langkah-langkah mengisih data tersebut dalam susunan menaik menggunakan isihan buih (bubble sort) dan Isihan Masukan (insertion sort)

5	1	4	2	8
---	---	---	---	---

Rajah 25

(10 markah)

- S26** Tuliskan satu aturcara lengkap C++ menggunakan fungsi rekursif untuk menghasilkan output seperti Rajah **S26**. Aturcara tersebut mestilah meminta pengguna untuk memasukkan bilangan baris dalam corak dan menggunakan fungsi rekursif untuk menghasilkan corak tersebut. Contohnya, memasukkan 4 sebagai input akan menghasilkan corak dalam Rajah **S26**.

*	*	*	*
*	*	*	
*	*		
*			
*			
*	*		
*	*	*	
*	*	*	*

Rajah S26

(20 markah)

**TERBUKA**

**-SOALAN TAMAT-**