UNIVERSITI TEKNOLOGI MALAYSIA | Sekolah Pendidikan Profesional dan Pendidikan Berterusan (SPACE)

TEST 1 / *UJIAN 1*
SEMESTER 2 – SESSION 2024/2025 / *SEMESTER 2 – SESI 2024/2025*

COURSE CODE *KOD KURSUS*	:	DSPD1733
COURSE NAME *NAMA KURSUS*	:	DATA STRUCTURES AND ALGORITHMS *STRUKTUR DATA DAN ALGORITMA*
YEAR / PROGRAMME *TAHUN / PROGRAM*	:	1 DSPD
DURATION *TEMPOH*	:	1 HOURS 30 MINUTES
DATE *TARIKH*	:	FEBRUARY 2025 *FEBRUARI 2025*

INSTRUCTION :
ARAHAN

Answer ALL Questions

Jawap SEMUA Soalan

(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 / PROGRAMME *TAHUN / PROGRAM*	:	.....
STUDENT'S SECTION *SEKSYEN*	:	.....
LECTURER'S NAME *NAMA PENSYARAH*	:	.....

This examination paper consists of …11… pages including the cover
Kertas soalan ini mengandungi …11…. muka surat termasuk kulit hadapan

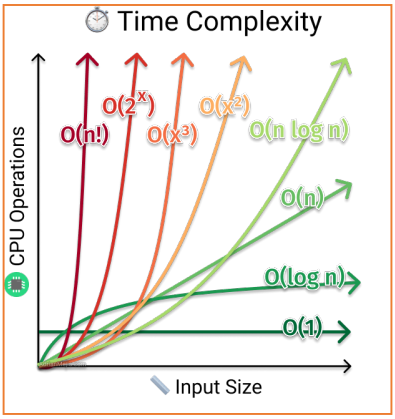
SECTION A: TRUE FALSE QUESTION [5 MARKS]

BAHAGIAN A: SOALAN BENAR SALAH [5 MARKAH]

1. The dataset size is one of the criteria to be measured when choosing the right data structure.
   Saiz set data adalah salah satu kriteria yang perlu diukur apabila memilih struktur data yang betul.

2. Best case, worse case and null case are all cases that exist during calculating time complexity.
   Kes terbaik, kes terburuk dan kes nol adalah kesemua kes yang wujud semasa mengira masa kerumitan.

3. Given algorithm A and B have a worst-case running time of O(n) and O(log n), respectively. Based on the graph below, it can be said that algorithm B always runs faster than algorithm A.
   Diberi algoritma A dan B mempunyai masing-masing kes terburuk masa larian adalah O(n) dan O(log n). Berdasarkan pada graf di bawah, boleh dikatakan bahawa algoritma B sentiasa berjalan lebih cepat daripada algoritma A.

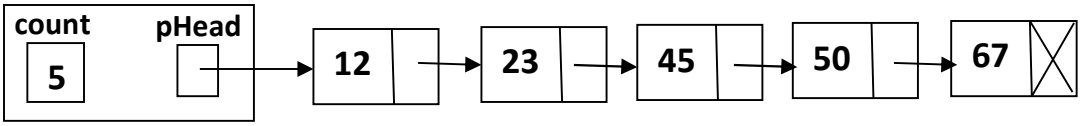


4. With the suitable use of data structure in the program, the memory used may be decreased.
   Dengan penggunaan struktur data yang sesuai dalam program, memori yang digunakan mungkin berkurangan.

5. Given below is a set of data to be inserted into a singly linked list sequentially.
   Di bawah ialah satu set data untuk dimasukkan ke dalam senarai yang dipautkan secara berurutan.

   45, 67, 23, 50, 12

   The singly linked list content will look like below.
   Kandungan senarai pautan tunggal akan kelihatan seperti di bawah.



Section A / *Bahagian A* **[5 M]**	
Question *Soalan*	Answer / *Jawapan* ***(TRUE / FALSE)***
1	
2	
3	
4	
5	

SECTION B: MULTIPLE CHOICE QUESTION [10 MARKS]

BAHAGIAN B: SOALAN ANEKA PILIHAN [10 MARKAH]

1. How is time complexity measured?

   Bagaimanakah masa kerumitan diukur?

   A. By counting the number of function in a program.
   Dengan mengira bilangan fungsi dalam satu aturcara.

   B. By counting the size of data input to the program.
   Dengan mengira saiz input data kepada aturcara.

   C. By counting the number of primitive operations performed by the program on a given input size.
   Dengan mengira bilangan operasi primitif yang dilakukan oleh aturcara pada saiz input yang diberikan.

   D. By counting the number of memory space being used for the program.
   Dengan mengira bilangan ruang memori yang digunakan untuk aturcara.

2. Which of the following points is not true about the singly linked list data structure when compared with list using array?
   Manakah antara perkara berikut adalah tidak benar tentang struktur data senarai bepaut tunggal apabila ia dibandingkan dengan senarai menggunakan tatasusunan?

   A. Direct random access is not allowed in a typical singly linked list implementation.
   Akses rawak secara langsung tidak dibenarkan dalam pelaksanaan senarai pautan tunggal biasa.

   B. Arrays have better cache locality that can make them better in terms of performance.
   Tatasusunan mempunyai lokasi cache yang lebih baik yang membolehkannya menjadi lebih baik dari segi prestasi.

   C. It is easy to insert and delete elements in singly linked list compare to using an array.
   Mudah untuk memasukkan dan memadam elemen dalam senarai pautan tunggal berbanding menggunakan tatasusunan.

   D. Access of elements in singly linked list takes less time than compared to arrays.
   Capaian elemen dalam senarai pautan tunggal mengambil masa yang lebih singkat berbanding dengan tatasusunan.

3. Consider the following algorithm snippet for inserting elements into a linked list. At what position will the new element be added to the list?

Pertimbangkan keratan algoritma berikut untuk memasukkan elemen dalam senarai terpaut. Pada kedudukan manakah elemen baharu akan ditambah pada senarai?

```
If (pPre null)
    pNew->link = list.head
    list.head  = pNew
```

A. Adding before first node / *Menambah sebelum nod pertama*

B. Adding after first node / *Menambah selepas nod pertama*

C. Adding in between two node / *Menambah di antara dua nod*

D. Adding as last node / *Menambah sebagai nod terakhir*

4. The postfix form of the infix expression below.

Bentuk postfix bagi ungkapan infiks di bawah.

(A+ B)*(C*D- E)*F / G

A. AB + CD * E - F * * G /

B. AB + CD * E - * F * G /

C. + AB * - * CDEF * / G

D. / * * + AB - * CDEFG

5. What is the value of the postfix expression below?

Apakah nilai ungkapan postfix di bawah?

4 3 6 3 * 12 - * +

A. - 14

B. 22

C. 94

D. 30

Section B / *Bahagian B* **[10 M]**	
Question *Soalan*	Answer / *Jawapan* **(A / B / C / D)**
1	
2	
3	
4	
5	

SECTION C: SUBJECTIVE QUESTION [20 MARKS]

BAHAGIAN C: SOALAN SUBJEKTIF [20 MARKAH]

QUESTION / *SOALAN* 1

By inspecting the pseudocode below, count the maximum number of primitive operations executed and estimate worst case running time complexity. **[10M]**

Dengan memeriksa pseudokod di bawah, kira bilangan maksimum operasi primitif yang dilaksanakan dan anggaran kerumitan masa larian kes terburuk.

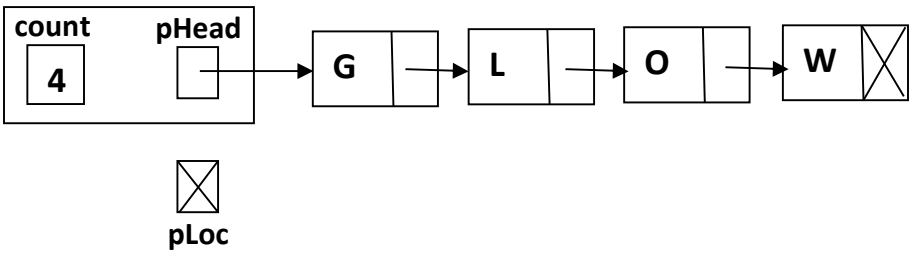
```
Algorithm accelerate(n)
begin
    for i⟵1 to n do
        print "increase speeding..."
        speed ⟵ speed * i
        if (speed equal limit)
            break
    {increment counter i * 2}
    end for
    print "have a nice day."
end
```

Answer / *Jawapan:*

QUESTION / *SOALAN* 2

Given the illustration of a linked list below,

Diberi ilustrasi senarai pautan tunggal seperti di bawah.



Where **pHead** and **pLoc** is pointer that will be pointing at a node (containing **data** and the **next** pointer). Write statement codes to change the last node's **next** in the linked list pointing to the first node. (Use **for** loop in your code) **[5M]**

*Di mana **pHead** dan **pLoc** ialah penunjuk yang akan menunjuk pada nod (mengandungi **data** dan penunjuk **next**). Tulis kod penyataan untuk menukar penuding **next** nod terakhir dalam senarai berpaut menunjuk ke nod pertama. (Guna gelung **for** di dalam kod anda)*

Answer / *Jawapan:*

QUESTION / *SOALAN* 3

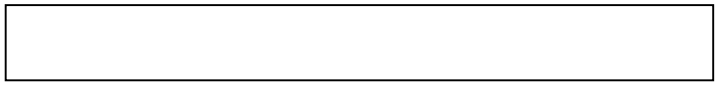
Imagine there are an empty integer linked list **ls** and two empty stacks of integers **s1** and **s2**. What is the content inside **ls**, **s1** and **s2** after statements below are executed? **[5M]**

*Bayangkan terdapat senarai berpaut integer yang kosong **ls** dan dua tindanan integer yang kosong **s1** dan **s2**. Apakah kandungan di dalam **ls**, **s1** dan **s2** selepas penyataan di bawah dilaksanakan?*

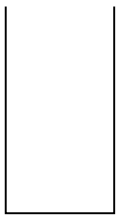
```
createLinkedList ls
createStack s1, s2
push(s1, 45)
push(s1, 28)
push(s1, 32)
push(s1, 67)
push(s1, 56)

while (!emptyStack(s1)
      pop(s1, x)
      if (x % 2 equal 0) then
          insert(ls, x)
      else
          push(s2, x)
end while
```

Answer / *Jawapan:*

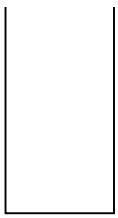
ls



s1



s2



SECTION D: PROGRAMMING QUESTION [15 MARKS]

BAHAGIAN D: SOALAN PENGATURCARAAN [15 MARKAH]

Write one program to cater all requirements below.

Given to you class declaration for Stack. Based on this declaration create the object and call the suitable member function to solve the given requirements.

Tulis satu program untuk memenuhi semua keperluan di bawah.

Diberikan kepada anda pengisytiharan kelas untuk Stack. Berdasarkan pengisytiharan ini cipta objek dan panggil fungsi ahli yang sesuai untuk menyelesaikan keperluan yang diberikan.

```
class Stack{
    private:
        NODE *pTop;
        int count;

    public:
        Stack();
        bool isEmpty();
        bool push(DATA dataIn);
        bool pop(DATA &dataOut);
        bool stackTop(DATA &dataTop);
        void display();
};
```

Read a list of numbers from the user. Push all the numbers into the stack. After completing pushing all the numbers into the stack, you need to pop until the stack is empty. During pop, please display the number and if the number is odd you need to add 1 to the number before display. The output of the program should look like being shown below.

Baca senarai nombor daripada pengguna. Tolak semua nombor ke dalam tindanan. Selepas selesai menolak semua nombor ke dalam tindanan, anda perlu pop sehingga tindanan kosong. Semasa pop, sila paparkan nombor dan jika nombor itu ganjil anda perlu menambah 1 pada nombor sebelum dipaparkan. Keluaran program sepatutnya kelihatan seperti yang ditunjukkan di bawah.

Output / *Keluaran*

```
How many number to read? 4
Enter number 1: 35
Enter number 2: 22
Enter number 3: 29
Enter number 4: 57

Result: 58  30  22  36
```

Answer / *Jawapan:*