UNIVERSITI TEKNOLOGI MARA
FINAL EXAMINATION

COURSE	:	**DATA STRUCTURES AND ALGORITHMS**
COURSE CODE	:	**ECE532**
EXAMINATION	:	**JUNE 2018**
TIME	:	**3 HOURS**

INSTRUCTIONS TO CANDIDATES

1. This question paper consists of five (5) questions.
2. Answer ALL questions in the Answer Booklet. Start each answer on a new page.
3. Do not bring any material into the examination room unless permission is given by the invigilator.
4. Please check to make sure that this examination pack consists of :
   i) the Question Paper
   ii) an Answer Booklet – provided by the Faculty
5. Answer ALL question in English.

DO NOT TURN THIS PAGE UNTIL YOU ARE TOLD TO DO SO

This examination paper consists of 10 printed pages

QUESTION 1

a) Dynamic data structures can grow and shrink during program executions.

i) Identify **TWO (2)** differences between linear and non-linear data structures.

(2 marks)

ii) Illustrate a **BINARY TREE** with integers values as given below with 10 as the root node.

10 25 5 12 2 7 24 8 55 11

(3 marks)

b) **Figure Q1b(i)** shows five nodes connected in a linked list.

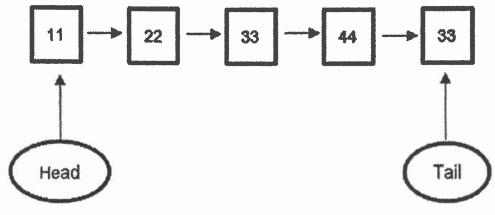


Figure Q1b(i)

i) Determine the content of each node shown in **Figure Q1b(i)** when the following programming codes are executed:

```
int main(){
  int i, temp, bilNode=5;
  for(i=0;i<(bilNode-1);i++)
  {
     struct node*A=Head;
     while(A->link!=NULL)
     {
         if(A->info < A->link->info)
         {
               temp=A->info;
               A->info=A->link->info;
               A->link->info=temp;
          }
     A=A->link;
     }
  }
return 0;}
```

Figure Q1b(ii)

(5 marks)

ii) Based on the following function prototype,

```
void insertAfter (node*P,int V);
```

write the function definition that will insert a new element at the back of the linked-list. (hint : `P` represents a pointer pointing to the last element of the list, `V` is the new element to be inserted at the back of the list)

(5 marks)

c) **Figure Q1c** demonstrates a program with pointers.

```
1  #include <iostream>
2  using namespace std;
3
4  int compare(const char *s1, const char *s2);
5
6  int main()
7  {
8      char string1[80];
9      char string2[80];
10     cout<<"Enter two strings: "<<endl;
11     cin>>string1;
12     cin>>string2;
13     cout<<"The result: "<< compare(string1, string2);
14     return 0;
15 }
16
17 int compare(const char *s1, const char *s2)
18 {
19     for( ; *s1 != '\0' && *s2 !='\0'; s1++, s2++)
20     {
21         if (*s1 != *s2)
22             return 0;}
23     }
24     return 1;
25 }
```

Figure Q1c

i) Explain the operations that occur in line **19** and **21**.

(4 marks)

ii) Indicate the output for the program in **Figure Q1c** if the input for `string1` and `string2` are "HAPPY" and "HOLIDAY" respectively.

(1 mark)

QUESTION 2

a) Based on the given structures declaration shown in **Figure Q2a:**

```
struct Info {
    char name[100];
    int age[3];
    string address;
};

struct Faculty {
    char dept[10];
    long ID;
    string location;
};
```

Figure Q2a

i) Declare the structure variables for `struct Info` and `struct Faculty`.

(2 marks)

ii) In the `main()` program, write C++ statements to get inputs from user for variables `name`, `age` and `dept`.

(3 marks)

b) **STACK** can be utilized to construct different type of expressions. By applying its concept, answer the following questions:

i) Analyze the following **PREFIX** expression, and determine its answer. Show the steps to get the final answer by using table.

/ + 8 / * 4 5 2 + 1 1

(5 marks)

ii) For the following expression, identify its **POSTFIX** equivalent by using conversion table.

A * B – C ^ (D / E)

(5 marks)

c) Based on the Binary Search Tree shown in **Figure Q2c**, identify and write the nodes sequence for the following search algorithms:

i) **Depth-First Search** (DFS) algorithm.
ii) **Breadth-First Search** (BFS) algorithm.

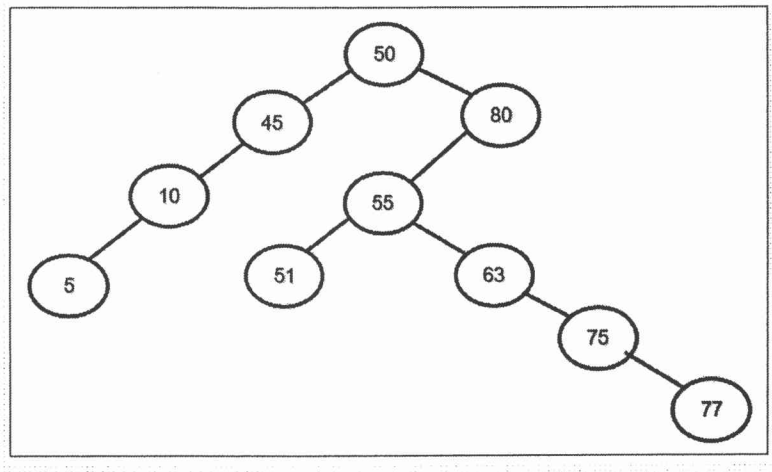


Figure Q2c

(5 marks)

QUESTION 3

a) A `main()` program that accepts an array as input is shown in **Figure Q3a**. Write the function definition for `int findMax(int *x, int y)`. This function will find the maximum value from array `x` with a size of `y`.

```
int main()
{
  int num;
  cout<<"Enter array size "<<endl;
  cin>>num;
  int arr[num];
  cout<<"Enter array values "<<endl;
  for(int i = 0; i<num; i++)
    cin>>arr[i];

  cout<<"The max number: "<<findMax(arr, num);
  return 0;
}
```

Figure Q3a

(5 marks)

b)
```
int numbers[10] = {1,2,3,4,5,6,7,8,9,10};
```

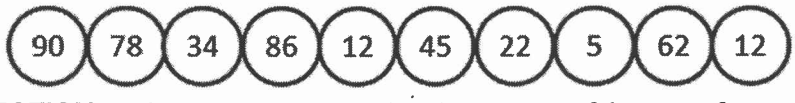
Based on the array declaration above, write the function definition for the following function prototype.

```
int findSum ( int [], int );
```

The function will calculate the sum of elements of the array by using **RECURSIVE** function concept.

(5 marks)

c) The following numbers were randomly selected:



By using **SELECTION** sorting concept, determine the content of the array for each iteration. You must clearly show the content of the array in each iteration.

(10 marks)

QUESTION 4

a) In data structures, stack and queue are very useful in computer applications.

i) Explain the concept applied for **STACK** and **QUEUE**.

(2 marks)

ii) Describe **TWO (2)** basic operations that can be performed by using **QUEUE**.

(2 marks)

iii) Suppose the following operations in **Figure Q4a** are performed on an empty stack:

```
int main()
{
  int x = 4;
  int y = 0;

  push(7);
  push(x++);
  pop();
  push(++y);
  pop();
  push(x*y);

  return 0;
}
```

Figure Q4a

Indicate the content of the static stack when the operations above have been executed. (Illustrate the process step by step).

(6 marks)

b)

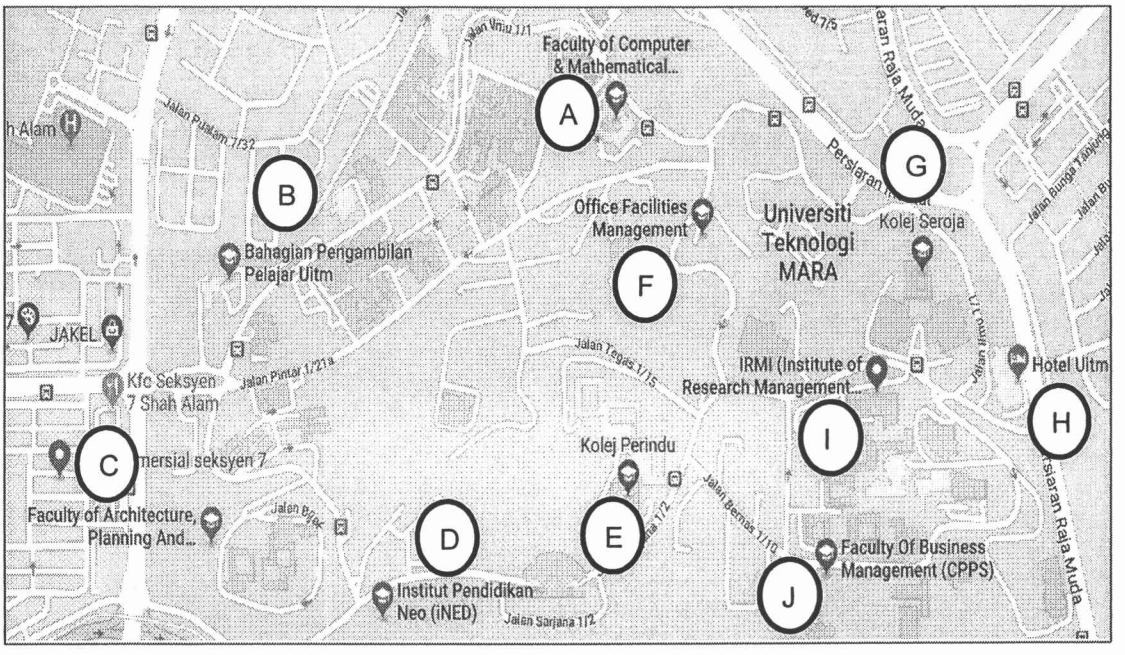


Figure Q4b

Figure Q4b shows ten selected locations in UiTM Shah Alam on a Google Map image. These locations are represented by ten different alphabets as indicated in the figure. The distance between two locations are the same back and forth (for example A to E and E to A have the same distance). The distances are shown in **Table Q4b**. Starting from location A, find the shortest path to all other locations by using Dijkstra's Algorithm.

From	To	Distance (km)
A	E	1.3
A	G	1.8
A	H	1.5
B	C	0.85
C	E	1.1
D	G	1.6
E	B	1.5
E	G	1.0
F	G	1.0
F	H	1.2
F	I	0.95
H	J	1.0
I	A	1.5
I	D	1.4

Table Q4b

(10 marks)

QUESTION 5

a) Determine and write the function definition for selection sorting algorithm using C++ programming codes based on the following function prototype:

```
void selectionSort(int Array[], int n);
//n is the number of element in the array
```

(5 marks)

b) A minimum spanning tree is a graph which consists of all vertices, and it may not contain all original edges.

i) Based on the graph shown in **Figure Q5b(i),** design the minimum spanning tree by using **Prim's Algorithms**, starting with node **A**.

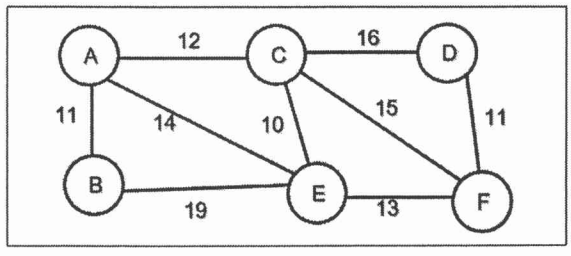


Figure Q5b(i)

(5 marks)

ii) Based on the graph shown in **Figure Q5b(ii),** design the minimum spanning tree and calculate the total minimum cost by using **Kruskal's Algorithms**.

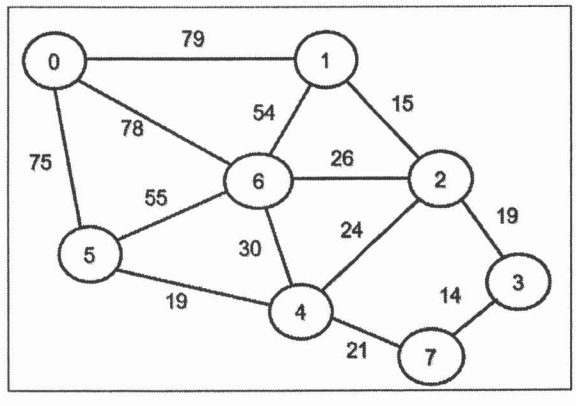


Figure Q5b(ii)

(5 marks)

c) Linear search or also known as sequential search is a method for finding a searched element within a list. It will sequentially check each element in the list one-by-one until a match has been found or until all elements have been checked.

Based on the following function prototype :

```
int linearSearch(int arr[], int n, int item);
```

write the function definition to perform a linear search, where `arr[]` is the set of numbers, `n` is the array size and `item` is the searched element. The function will return value **1** if the searched element is found. Otherwise, **0** will be returned.

(5 marks)

END OF QUESTION PAPER