UTHM
Universiti Tun Hussein Onn Malaysia

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER II
SESSION 2022/2023

COURSE NAME	:	DATA STRUCTURE
COURSE CODE	:	BIC 10404
PROGRAMME CODE	:	BIP / BIS / BIM / BIW
EXAMINATION DATE	:	JULY / AUGUST 2023
DURATION	:	3 HOURS

INSTRUCTION

1. ANSWER ALL QUESTIONS.
2. THIS FINAL EXAMINATION IS CONDUCTED VIA **CLOSED BOOK**.
3. STUDENTS ARE **PROHIBITED** TO CONSULT THEIR OWN MATERIAL OR ANY EXTERNAL RESOURCES DURING THE EXAMINATION CONDUCTED VIA CLOSED BOOK.

THIS QUESTION PAPER CONSISTS OF **SIX (6)** PAGES

Q1 Determine whether each of the following statements is **TRUE** or **FALSE** about graph.

(a) Two standard ways of searching for a graph are depth-first search and breadth-first search. Both searching algorithms may begin at a designated vertex, but they differ with respect to the order in which they visit the remaining vertices.

(2 marks)

(b) A graph may be represented using both, array and linked list.

(2 marks)

(c) In a directed graph, some edges may not be assigned directions, therefore it is not the same as a multigraph.

(2 marks)

(d) A simple path is a path such that all vertices are distinct, except that the first and last could be the same.

(2 marks)

(e) A graph is said to be complete if at least one node in the graph is adjacent to another node.

(2 marks)

Q2 Write basic steps for implementing Breadth-First Search (BFS) algorithm for graph traversal in C programming.

(10 marks)

Q3 Answer **Q3(a) to Q3(d)** based on **Figure Q3**.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define SIZE 10

struct staff {
    char name[20];
    double salary;
}

typedef struct staff Staff;

struct StaffNode {
    staff staffData;
    struct StaffNode *next;      }

int main() {
    int i;
    char staffName[20];
    double staffSalary;

    struct StaffNode *head = NULL, *q, *tail;
    tail = head;

    for(i=0; i<SIZE ; i++)
    {
      //queue functions.
    }
```

Figure Q3

(a) Write algorithm function for `empty( )`.

(5 marks)

(b) Write algorithm for `dequeue( )`.

(9 marks)

(c) Describe the meaning of Underflow queue.

(3 marks)

(d) Describe the meaning of Overflow queue.

(3 marks)

Q4 Answer **Q4(a) to Q4(d)** based on **Figure Q4.**

```
#include <stdio.h>
#define TRUE 1
#define FALSE 0
#define SIZE 6

int searching1(int n[],int size,int data);
{
   int idx;
   for(idx=0; idx<size; idx++)
   {
     //1- if element in the current index equal to target data
     //2- return the index
   }
   return -1;
}
```

```
int searching2(int n[],int size, int data)
{
   int index, found, start, end, mid;
   index = -1;
   found = FALSE;
   start=0;
   end = size-1;

   while((start<=end) && !found)
   {
     //3-set mid to (start+end)/2
     //4-if target data equal to n[mid] then
     //5- set found to TRUE
     //6- set index to mid
     //7-otherwise if (target data > n[mid] then
     //8- set start equal to mid + 1

   else
           end = mid-1;
    }
      return(index);
}

int main()
{
  int n[SIZE]= {18,45,90,64,120,87};
  int item = 64, searchingData1, searchingData2;
  //9- searchingData1 = function calling to searching1() function
  //10-searchingData2 = function calling to searching2() function
  return 0;
}
```

Figure Q4

(a) Write the code segments of the algorithm for `searching1()` function in the comment section labelled from 1 to 2 in C programming.

(3 marks)

(b) Write the code segments of the algorithm for `searching2()` function in the comment section labelled from 3 to 8 in C programming.

(6 marks)

(c) Write the appropriate function calling in `main()` for `searching1()` and `searching2()` functions in the comment section labelled from 9 to 10 in C programming.

(2 marks)

Q5 Answer **Q5(a) to Q5(d)** based on the information given in **Figure Q5.**

```
void sortAlgo(int size,int data[]){
 int curr=1,temp,x;

 while(curr < size)
 {
  for(x=0;x < size-curr;x++){
  if(data[x] > data[x+1])    {
    // 1- Set temp to data[x]

    // 2-Set data[x+1] to data[x]

   // 3-Set data[x+1] to temp;
   }
  }

   printf("\n\n Data = %d \n ", curr);
  for(x=0;x < size;x++)
    // 4- Display data printf(" %d ",data[x]);
  curr++;
 }
}
```

Figure Q5

(a) Name the type of sorting algorithm defined in `sortAlgo()` function.

(1 mark)

(b) Write the code segment of the algorithm for `sortAlgo()` function in the comment section labelled from 1 to 3 in C programming.

(6 marks)

(c) Write the code segment to display the data in `sortAlgo()function` as in the comment labelled with 4 in C programming.

(2 marks)

Q6 Answer **Q6(a) to Q6(e)** based on Node represented in **Figure Q6.**

```
struct node {
     int data;
     struct node *left;
     struct node *right;
};
```

Figure Q6

(a) Write the method code so it returns the `sum of the values` contained in all the nodes of the binary tree with root `n` in C programming.

(8 marks)

(b) Draw/List the tree in **Preorder** traversal way that is created if the following numbers are inserted in the tree in the given order.

12 15 3 35 37 20 50 23

(4 marks)

(c) Draw the tree in **Inorder** traversal way that is created if the following numbers are inserted in the tree in the given order.

30 6 70 14 60 10 5 6

(4 marks)

(d) Draw the tree after deleting 10 in **Inorder** traversal way based on answer in **Q6(c)**.

(4 marks)

-END OF QUESTIONS –