



UNIVERSITI TUN HUSSEIN ONN MALAYSIA

**FINAL EXAMINATION
(ONLINE)
SEMESTER II
SESSION 2019/2020**

COURSE NAME	:	DATA STRUCTURE AND ALGORITHM
COURSE CODE	:	BIT 10703
PROGRAMME CODE	:	BIT
EXAMINATION DATE	:	JULY 2020
DURATION	:	3 HOURS
INSTRUCTION	:	1. ANSWER ALL QUESTIONS. 2. PLEASE MAKE SURE TO CLICK "SAVE ANSWER" BUTTON FOR SUBJECTIVE QUESTIONS. OBJECTIVE QUESTIONS ARE SAVED AUTOMATICALLY.

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

Q1 Determine the output for the program in **Figure Q1**.

```
#include <stdio.h>

long testNum( long n );
int main()
{
    long result;

    result = testNum (5);
    printf( "\nResult = %ld\n", result );

    return 0,
}

long testNum( long n )
{
    if ( n == 0 || n == 1 ) {
        printf(" %ld", n);
        return n;
    }
    else {
        return testNum( n - 1 ) + testNum( n - 2 );
    }
}
```

FIGURE Q1

(5 marks)

Q2 Answer **Q2(a)** and **Q2(b)** based on the program given in **Figure Q2**.

```
#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, *p1;

    for(i=0; i<SIZE; i++)
    {
        printf("\nEnter name:");
        scanf("%s", staffName);
        printf("\nEnter salary:");
        scanf("%lf", &staffSalary);
```

```

    p1 = malloc(sizeof(struct StaffNode));

    strcpy(p1->staffData.name, staffName);
    p1->staffData.salary = staffSalary;
    p1->next = NULL;

    if (head == NULL)
        head = p1;
    else
        { p1->next = head,
          head = p1;}
}

return 0,
}

```

FIGURE Q2

- (a) Write a function, named `findMax` for **Figure Q2** that will return the maximum salary from the linked list. (10 marks)
- (b) Write a code segment that will remove a node from the linked list according to stack operation. (10 marks)

Q3 Complete **Figure Q3** with a code segment that will insert a new node to the linked list according to queue operation.

```

#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, *p1;

    for(i=0; i<SIZE; i++)
    {
        printf("\nEnter name:");
        scanf("%s", staffName);
        printf("\nEnter salary:");
    }
}

```

TERBUKA

```

scanf("%lf",&staffSalary);

p1 = malloc(sizeof(struct StaffNode));

strcpy(p1->staffData.name,staffName);
p1->staffData.salary = staffSalary;
p1->next = NULL;

Answer:

} //for
return 0;
}

```

FIGURE Q3

(10 marks)

- Q4** (a) **Figure Q4(a)** presents a program with a sorting function, called SortData. Determine the output for **Figure Q4(a)**.

```

#include <stdio.h>
void SortData( int list[], int no)
{
    int i, j, largest, largestidx, temp, k;

    for (i = 0; i <no-1; i++)
    { largest = list[i];
      largestidx = i;
      for (j = i+1; j <no; j++)
          if (list[j] > largest) {
              largest = list[j];
              largestidx = j;
          }

      if (largest>list[i]){
          temp = list[i];
          list[i] = largest;
          list[largestidx] = temp;
      }
      printf("\nPass %d:", i+1);
      for (k = 0, k<no; k++)
          printf("%5d", list[k]);
    }
}

int main()
{
    int list[ ] = {25, 57, 48, 37, 12, 65, 98, 75, 83};
    SortData(list,9);

    return 0;}

```

FIGURE Q4 (a)

(8 marks)

TERBUKA
CONFIDENTIAL

(b) Name the sorting algorithm performed in **Q4 (a)**.

(2 marks)

(c) Determine the output for **Figure Q4 (b)**.

```
int testQ3( int list[], int no, int x)
{
    int idx, test, left, right, mid;

    idx = -1;
    test = 0;

    left = 0;
    right = no - 1;

    while (left <= right && !test)
    {
        mid = (int)((left+right)/2);
        if (x == list[mid]){
            test = 1;
            idx = mid;
        }
        else if (x > list[mid])
            left = mid + 1;
        else
            right = mid - 1;
    }
    return idx;
}

int main()
{
    int i, list[10];
    for(i=0; i<10; i++)
        list[i] = i+3*i;

    printf("\nOutput: %d ", testQ3(list, 10, 36));
    printf("\nOutput: %d ", testQ3(list, 10, 8));
    printf("\nOutput: %d ", testQ3(list, 10, 40));
    printf("\nOutput: %d ", testQ3(list, 10, 20));
    printf("\nOutput: %d ", testQ3(list, 10, 3));
    return 0;
}
```

FIGURE Q4(b)

(10 marks)

Q5 Answer 5(a) - 5(b) using the sequence of values given in **Figure Q5**.

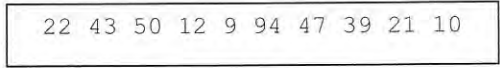


FIGURE Q5

(a) Draw a binary search tree for the sequence of values given in **Figure Q5**.
(10 marks)

(b) Determine the results of each traversal algorithm for the answer in **Q5(a)**

Traversal Algorithm	Auswet
Preorder	
Inorder	
Postorder	

(15 marks)

- END OF QUESTION-

