# UTHM
Universiti Tun Hussein Onn Malaysia

# UNIVERSITI TUN HUSSEIN ONN MALAYSIA

# FINAL EXAMINATION
## SEMESTER II
## SESSION 2017/2018

| | | |
|---|---|---|
| COURSE NAME | : | DATA STRUCTURE |
| COURSE CODE | : | BIC 10404 |
| PROGRAMME CODE | : | BIC |
| EXAMINATION DATE | : | JUNE / JULY 2018 |
| DURATION | : | 3 HOURS |
| INSTRUCTIONS | : | A) ANSWER **ALL** QUESTIONS |
| | | B) PLEASE WRITE YOUR ANSWERS IN THIS QUESTION BOOKLET |

TERBUKA

THIS QUESTION PAPER CONSISTS OF **ELEVEN (11)** PAGES

Q1    Answer **Q3(a)** – **Q3(c)** based on **Table 3**. It shows montly kerepek production (in kg) for Thunder Sdn. Bhd. in year 2017.

**Table 3.**

| Month | Number of Units |
|---|---|
| January | 35 |
| February | 25 |
| March | 89 |
| April | 90 |
| May | 55 |
| June | 78 |
| July | 28 |
| August | 12 |
| September | 45 |
| October | 20 |
| November | 30 |
| December | 37 |

(a)    Write a full program code to perform `bubble sort algorithm` for the kerepek production.

(15 marks)

**Answer:**

**CONFIDENTIAL**

(b)     Show the sequence of production in every pass of selection sort algorithm.

(12 marks)

**Answer:**

**CONFIDENTIAL**

(c)     Name **THREE (3)** other sorting algorithms.

(3 marks)

| | Answer |
|---|---|
| 1 | |
| 2 | |
| 3 | |

**Q2**     A function call to perform a linear search is included in the main program of **Figure Q2**. If a value is found, the linear search function shall return the position of the value in the array. Otherwise, -1 is returned from the function. Complete **Figure Q2** with implementation of the `linear search algorithm`.

(10 marks)

```c
#include <stdio.h>
#define SIZE 7
#define TRUE 1
#define FALSE 0

int main()
{
 int nums[SIZE] = {155, 60, 272, 820, 95, 217, 153};
 int position, num;
 int linearSearch(int[],int,int);

 printf("\nEnter a value that you want to search: ");
 scanf("%d",&num);
 position = linearSearch(nums,SIZE,num);

 if (position>0)
   printf("\nThe value is found at index %d",position);
 else
   printf("\nThe value is not found.");

 return 0;
}

int linearSearch(int list[],int size,int value)
{
 int i = 0, index = -1, found = FALSE;
```
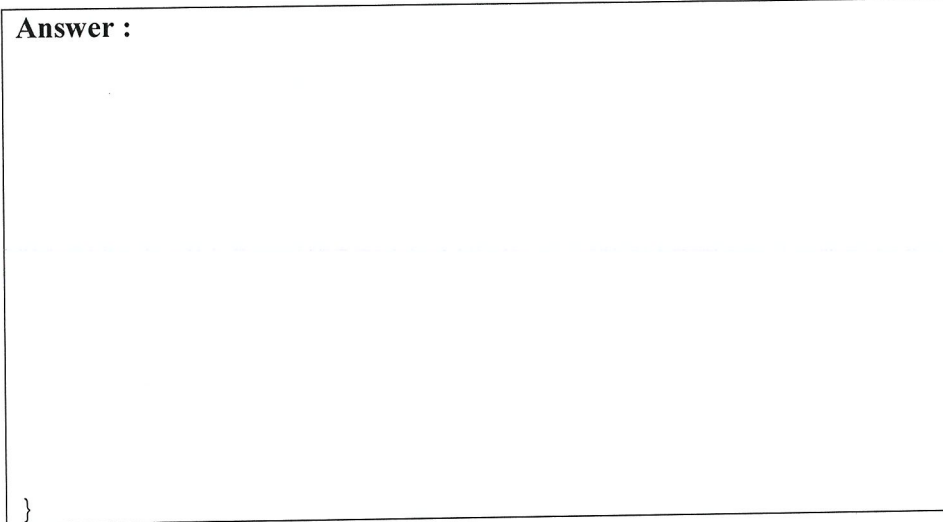
**Answer :**

```
}
```

**Figure Q2**

**Q3**    Answer **Q3(a)** and **Q3(b)** based on the information given in **Figure Q3**.

```c
#include <stdio.h>
#include <string.h>

struct EmpList{
        char name[20];
        double salary;
        struct EmpList *next;};

typedef struct EmpList EmpList;

int main()
{
 EmpList *front = NULL;
 EmpList e1, e2, e3, e4, e5;

 strcpy (e1.name,"Adila");
 strcpy (e2.name,"Azah");
 strcpy (e3.name,"Firdaus");
 strcpy (e4.name,"Hanani");
 strcpy (e5.name,"Rozlini");

 e1.salary = 4200.50;
 e2.salary = 2700.00;
 e3.salary = 5000.20;
 e4.salary = 3200.75;
 e5.salary = 4000.25;

 return 0;}
```

**Figure Q3**

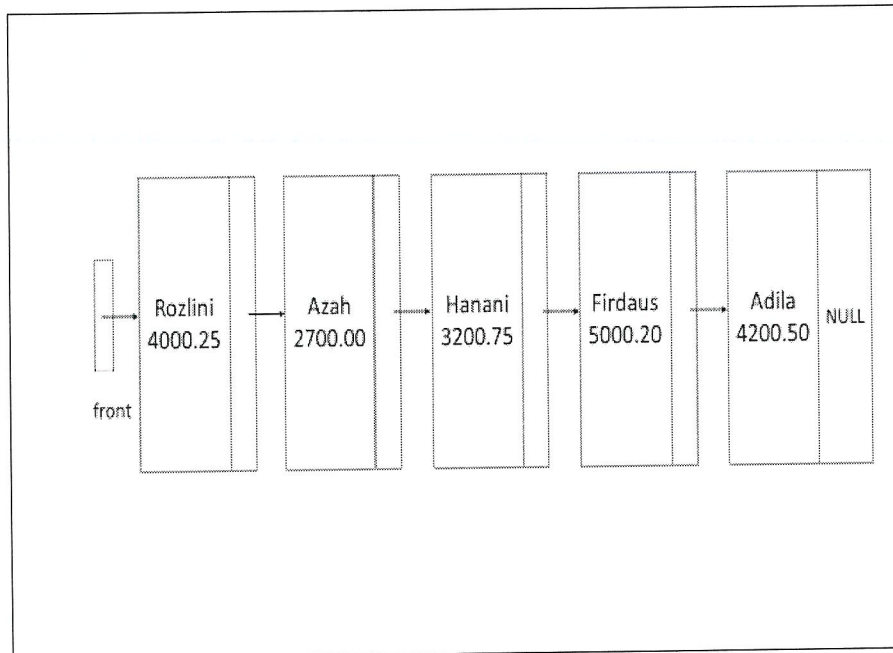(a)     Write code segment to establish the linked list in **Figure Q3(a)**.



**Figure Q3(a)**

(10 marks)

**Answer:**

(b)    Write a code segment to determine and display information of the employee with minimum salary from the linked list established in **Figure Q3(a)**.

(15 marks)

**Answer:**

**Q4**    Answer **Q4(a)** and **Q4(b)** based on the information given in **Figure Q4**.
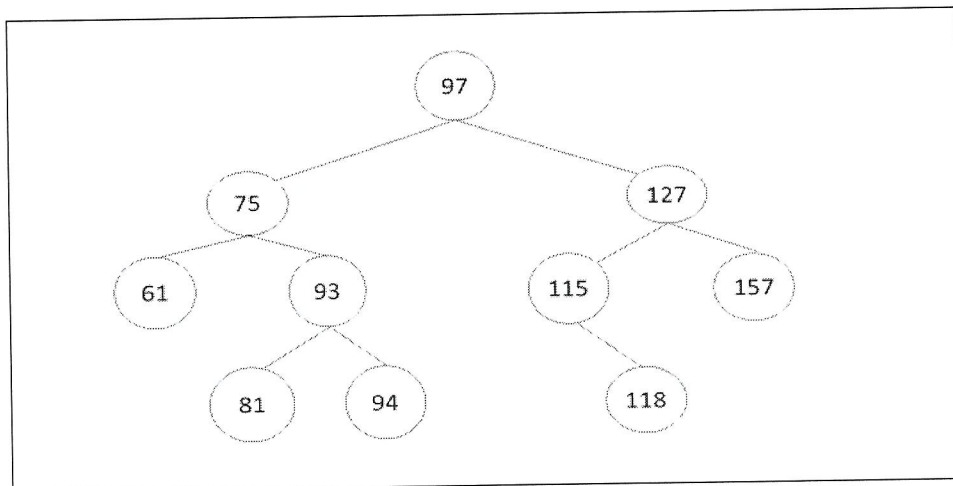


**Figure Q4**

(a)     Perform traversal algorithms for the tree in **Figure Q4**.

(15 marks)

| Traversal algorithms | Answer |
|---|---|
| Preorder | |
| Inorder | |
| Postorder | |

(b)     What is the value stored in root node?

(2 marks)

**Answer:**

**Q5**     Answer **Q5(a) – Q5(c)** based on the information given in **Figure Q5(a)** and **Figure Q5(b)**.

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

struct NumList{
        double num;
        struct NumList *next;};

typedef struct NumList NumList;

int main()
{
 NumList *head = NULL, *p1, *p2;
 double value, numArray[]={100.4,50.7,87.9,95.2,29.3};
 int i;
```

```
    for (i=0; i<5; i++){
      value = numArray[i];

      p1 = malloc(sizeof(NumList));

      if (p1!=NULL){
        p1->num = value*100;
        p1->next = NULL;}

      if (head == NULL)
        head = p1;
      else{
        p2 = head;
        while (p2 ->next!=NULL)
          p2 = p2->next;

        if (p2->next == NULL)
          p2->next = p1;}
    }

  p2 = head;
  head = head->next;
  free(p2);

  p2 = head;
  while (p2!=NULL){
    printf("%.2lf  ", p2->num);
    p2 = p2->next;}


  return 0;
}
```

**Figure Q5(a)**

(a)    Determine the output for **Figure Q5(a)**.

(9 marks)

**Answer:**

**CONFIDENTIAL**

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

struct NumList{
        double num;
        struct NumList *next;};

typedef struct NumList NumList;

int main()
{
 NumList *head = NULL, *p1, *p2;
 double value, numArray[]={100 4,50.7,87.9,95.2,29.3};
 int i;

 for (i=0; i<5; i++){
   value = numArray[i];

   p1 = malloc(sizeof(NumList));

   if (p1!=NULL){
     p1->num = value*10;
     p1->next = NULL;}

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

 p2 = head;
 head = head->next;
 free(p2);

 p2 = head;
 while (p2!=NULL){
     printf("%.2lf  ", p2->num);
     p2 = p2->next;}


 return 0;
}
```

**Figure Q5(b)**

**CONFIDENTIAL**

(b) Determine the output for **Figure Q5(b)**.

(9 marks)

**Answer:**

(c) Name the data structure concept that **BEST** describes each of the algorithm.

(2 marks)

| Algorithm | Answer |
|---|---|
| **Figure Q5(a)** | |
| **Figure Q5(b)** | |

- **END OF QUESTION** –