



**UNIVERSITI TUN HUSSEIN ONN MALAYSIA**

**PEPERIKSAAN AKHIR  
SEMESTER II  
SESI 2011/2012**

**NAMA KURSUS : STRUKTUR DATA**  
**KOD KURSUS : DAT 20104/DIT 2014**  
**PROGRAM : 2 DAT/2 DIT**  
**TARIKH PEPERIKSAAN : MAC 2012**  
**JANGKA MASA : 2 ½ JAM**  
**ARAHAN : JAWAB SEMUA SOALAN**

**KERTAS SOALAN INI MENGANDUNGI SEMBILAN (9) MUKA SURAT**

**SOALAN DI DALAM BAHASA MELAYU**

- S1** (a) Terangkan konsep struktur data primitif.  
(4 markah)
- (b) Nyatakan pengaturcaraan berstruktur.  
(5 markah)
- S2** (a) Tuliskan satu algoritma untuk memasukkan nod di awal senarai berpaut.  
(7 markah)
- (b) Tuliskan algoritma untuk memasukkan p nod pada akhir senarai berpaut.  
(5 markah)
- S3** (a) Tuliskan **EMPAT (4)** aplikasi timbunan.  
(4 markah)
- (b) Tuliskan **EMPAT (4)** aplikasi sistem gilir.  
(4 markah)
- S4** Tafsirkan keratan kod bagi Aturcara 1 dengan bantuan gambarajah yang sesuai.
- ```
Node *temp;  
temp = start_ptr;
```
- Aturcara 1  
(3 markah)

- S5** Tafsirkan keratan kod bagi Aturcara 2 dengan bantuan gambarajah yang sesuai.

```
start_ptr = start_ptr->next;
```

Aturcara 2

(2 markah)

- S6** Tafsirkan keratan kod bagi Aturcara 3 dengan bantuan gambarajah yang sesuai.

```
if (current == start_ptr)
    cout << "You are at the start of the list" << endl;
else
    {
        node *previous;
        previous = start_ptr;
        while (previous->next != current)
            previous = previous->next;
        current = previous;
    }
```

Aturcara 3

(5 markah)

- S7** Tuliskan algoritma bagaimana timbunan digunakan untuk mengesan sepasang pendakap kerinting, "}" dalam pengaturcaraan c + + . Bagaimana timbunan boleh digunakan untuk memeriksa sama ada ungkapan yang betul *parenthized* atau tidak serta menentukan sama ada ungkapan infiks mempunyai kurungan seimbang atau tidak.

(7 markah)

- S8** Pertimbangkan 8 nombor 50, 33, 44, 22, 77, 35, 60 dan 40. Paparkan pembinaan binari dengan memasukkan nombor tersebut dalam turutan yang diberikan.

(6 markah)

**S9** Tafsirkan keratan kod bagi Aturcara 4 dengan bantuan gambarajah yang sesuai.

```
void pop()
{
    node *temp1,*temp2;
    if(start_ptr==NULL)
        cout<<"The list is empty"<< endl;
    else
    {
        temp1=start_ptr;
        temp2=temp1;
        while(temp1->nxt!=NULL)
        {
            temp2=temp1;
            temp1=temp1->nxt;
        }
        if(temp1==temp2)
        {
            cout<<temp1->name<<" "<< endl;;
            cout<<temp1->department<<" "<< endl;;
            cout<<temp1->company<< endl;
            start_ptr=NULL;
        }
        else
        {
            cout<<temp1->name<<" "<< endl;;
            cout<<temp1->department<<" "<< endl;;
            cout<<temp1->company<< endl;;
            temp2->nxt=NULL;
            delete temp1;
        }
    }
}
```

**Aturcara 4**

(10 markah)

**S10** Tunjukkan fasa-fasa isihan jenis buih ke atas senarai 11, 15, 2, 13, 6

(8 markah)

**S11** Diberi senarai nod-nod inorder dan preorder. Bina sebuah pohon binari dari nod-nod diberikan seperti berikut:

Inorder : 10, 15, 17, 18, 20, 25, 30, 35, 38, 40, 50

Preorder: 20, 15, 10, 18, 17, 30, 25, 40, 35, 38, 50

(10 markah)

**QUESTION IN ENGLISH**

**Q1** (a) Explain the concept of primitive data structures. (4 marks)

(b) Define structured programming. (5 marks)

**Q2** (a) Write an algorithm to insert a node in the beginning of the linked list. (7 marks)

(b) Write an algorithm to insert a node p at the end of a linked list. (5 marks)

**Q3** (a) Write any **FOUR(4)** application of a stack (4 marks)

(b) Write any **FOUR (4)** applications of queues. (4 marks)

**Q4** Interpret part of Program 1 coding using appropriate diagrams.

```
Node *temp;  
temp = start_ptr;
```

Program 1

(3 marks)

**Q5** Interpret part of Program 2 coding using appropriate diagrams.

```
start_ptr = start_ptr->next;
```

Program 2

(2 marks)

**Q6** Interpret part of Program 3 coding using appropriate diagrams.

```
if (current == start_ptr)
    cout << "You are at the start of the list" << endl;
else
    {
        node *previous;
        previous = start_ptr;
        while (previous->next != current)
            previous = previous->next;
        current = previous;
    }
```

Program 3

(5 marks)

**Q7** Write an algorithm how stack is used to detect pair of curly braces, “}” in c++ programming. How can stacks be used to check whether an expression is correctly parenthesized or not. Using stacks, write an algorithm to determine whether the infix expression has balanced parenthesis or not.

(7 marks)

**Q8** Consider the following eight numbers 50, 33, 44, 22, 77, 35, 60 and 40. Display the construction of the binary by inserting the above numbers in the given order.

(6 marks)

**Q9** Interpret part of Program 4 coding using appropriate diagrams.

```
void pop()
{
    node *temp1,*temp2;
    if(start_ptr==NULL)
        cout<<"The list is empty"<< endl;
    else
    {
        temp1=start_ptr;
        temp2=temp1;
        while(temp1->nxt!=NULL)
        {
            temp2=temp1;
            temp1=temp1->nxt;
        }
        if(temp1==temp2)
        {
            cout<<temp1->name<<","<< endl;;
            cout<<temp1->department<<","<< endl;;
            cout<<temp1->company<< endl;
            start_ptr=NULL;
        }
        else
        {
            cout<<temp1->name<<","<< endl;;
            cout<<temp1->department<<","<< endl;;
            cout<<temp1->company<< endl;;
            temp2->nxt=NULL;
            delete temp1;
        }
    }
}
```

Program 4

(10 marks)



**Q10** Show the various passes of bubble sort on an unsorted list 11, 15, 2, 13, 6

(8 marks)

**Q11** Construct a binary tree whose nodes in inorder and preorder are given as follows:

Inorder : 10, 15, 17, 18, 20, 25, 30, 35, 38, 40, 50

Preorder: 20, 15, 10, 18, 17, 30, 25, 40, 35, 38, 50

(10 marks)