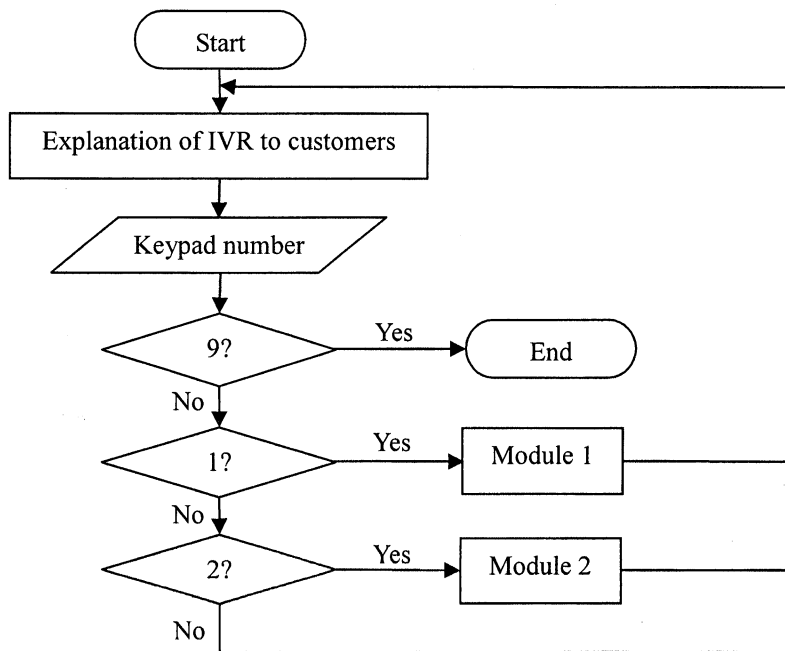


1. (a) (i) Implementation 1
 (ii) Integration 1
 (iii) A system test is done by developers while a UAT involves ultimate users to test the system with authentic data. (people & data type) 1
 (iv) Review and refine the previous process. 1
 (b) (i) 3 1
 (ii) 3 3



- (c) (i) The service can be based on the first-in-first-out policy. 1
 (ii) **[Pascal version]** ① **[Visual Basic version]** ①
 $Qfirst = Qlast$ ① $Qfirst = Qlast$ ①
 $Q[Qfirst]$ ① $Q[Qfirst]$ ①
 $(Qfirst + 1) \bmod n$ ①, ① $(Qfirst + 1) \bmod n$ ①, ①
 $+1 \bmod n$ ①
[C version] **[Java version]**
 $Qfirst == Qlast$ ① $Qfirst == Qlast$ ①
 $Q[Qfirst]$ ① $Q[Qfirst]$ ①
 $(Qfirst + 1) \% n$ ①, ① $(Qfirst + 1) \% n$ ①, ①
 (iii) The queue is empty. 1
 (d) (i) $n - 1$ 1
 (ii) Some record cannot be retrieved using `POP()`. 1
 / Some elements in the queue will be overwritten.

2. (a) (i)

2	false	false	true	false	true	false
---	-------	-------	------	-------	------	-------

2
- (ii) n^2 or 36 1
- (b) (i) The values they store are the same. 2
- (ii) When $i=j$, he does not need to store the information as it must be false. 1
 When $j<i$, the (j,i) -th element in A is the same as the (i,j) -th element with $i<j$, so these entries need not to be stored. 1
 The total number of entries with $i<j$ is 15.
- (c) (i) (1,5) (2,4) (2,6) (3,5) (4,6) (and their reverses) 2
- (ii) false (or 0) ① 3
 $\left. \begin{matrix} p, i \\ p, j \end{matrix} \right\}$ or their reverses; i and j are interchangeable ②
- (d) The development of mobile applications requires a short development life cycle. Library functions in object-oriented programming can help shorten the development life cycle. (reusability) 2

3. (a) (i) Amy, Jade 2
- (ii) any value larger than the array size / negative values 1
- (iii) 4 1

(b) 4

Address	Content	Previous	Next	
0	START	-1	3	
1	John	3	4	← negligible
2	Mary	4	-1	
3	Susan	0	4	
4	Fiona	3	2	

- ① each row
- (c) Advantage: The linked list can be transverse in two ways. 1
 Disadvantage: It requires more storage. 1
- (d) (i) Candy, Amy, Daisy 3
- (ii) It saves storage space. 1

	Marks
4. (a) (i) FALSE TRUE TRUE	3
(ii) It checks whether the string in ST is a palindrome (回文) or not.	1
(iii) for i from <u>1</u> to <u>n/2</u>	1
(b) FALSE	1
0	1
MyLen(T2)	1
i+j-1 (i+j ①)	2
(c) - Declaration	6
- Initialisation of a flag (FOUND) / maximum length	
- Any loop: 1 to length of T2 (n2)	
- Check all possible substrings in T2 (looping for '1+2+3+...+n2' checking)	
- Use of IsSub without correct parameters	
- All correct	
(① @)	

[Pascal version]

```

procedure LongSub(T1, T2 : string);
var i, j : integer;
    found : Boolean;
begin
    found := false;
    i := n2;
    while not found and (i >= 1) do begin
        j := n2 - i + 1;
        while not found and (j >= 1) do begin
            if IsSub(T1, MyCopy(T2, j, i)) then
                begin
                    found := TRUE;
                    writeln('The length is ', i);
                end;
            j := j - 1;
        end;
        i := i - 1;
    end;
end;

```

MyLen(MyCopy(T2, j, i))

[C version]

```

void LongSub( char T1[], char T2[]) {
    int i, j, found;
    char temp[50];
    found = 0;
    i = n2;
    while (!found && i >= 1) {
        j = n2 - i + 1;
        while (!found && j >= 1) {
            MyCopy(T2, temp, j, i);
            if (IsSub(T1, temp)) {
                found = 1;
                printf("The length is %i\n", i);
            }
            j--;
        }
        i--;
    }
}

```

[Visual Basic version]

```

Private Sub longSub(T1 As String, T2 As String)
    Dim i As Integer
    Dim j As Integer
    Dim found As Boolean
    found = False
    i = n2
    Do While (Not found And (i >= 1))
        j = n2 - i + 1
        Do While (Not found And (j >= 1))
            If IsSub(T1, MyCopy(T2, j, i)) Then
                found = True
                msg = msg & "The length is " & i & vbNewLine
            End If
            j = j - 1
        Loop
        i = i - 1
    Loop
End Sub

```

[JAVA version]

```

private static void LongSub( String T1, String T2) {
    int i, j;
    boolean found = false;
    i = n2;
    while (!found && i >= 1) {
        j = n2 - i;
        while (!found && j >= 0) {
            if (IsSub(T1, MyCopy(T2, j, j+i))) {
                found = true;
                System.out.println(j+" The length is "+i);
            }
            j--;
        }
        i--;
    }
}

```