KING SAUD UNIVERSITY

COLLEGE OF COMPUTER & INFORMATION SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

CSC212 Data Structures
Second Semester 1425/1426 AH
Second Mid-term Examination: Wednesday 10.04.1426 A.H./18.05.2005 C.E.
Duration: 2 Hours
Instructors: Dr. Aqil Azmi and Mr. Saad Al-Ahmadi

---

1. [Marks 15]

Consider the Ordered List (OL) ADT specifications, write a client function:

```
double meadian(OL<double> &ol);
```

Precondition: the ordered list is not empty.

Process: returns the median (الوسيط) of an ordered list of doubles. The median is defined only for sorted elements. If the number of elements is odd, the median is the element in the middle; otherwise it is the average of the two middle elements.

2. [Marks 15]

Assume that we are using pointers (single linked list) to implement the Ring:

```
template <class T>
class Ring {
      private:
            RingNode<T> *head, *current;
            int size;
      public: ......
};
```

Implement the member function:

```
template <class T>
int Ring<T>::insert(const T &x);
```

Consult the ADT specification sheet for the *precondition* and the *process*.

3. [Marks 6+6+6+7=25]

Assuming that *key* is denoted by $a_1 a_2 \ldots a_n$ where a_k is the k-th character in the word (a_n is the last character). Then the hash function $H(key)$ is given by,

$$H(key) = \left\{ n^2 + \sum_{i=1}^{2} \text{code}(a_i) \cdot \text{code}(a_{i+1}) \right\} \bmod 9.$$

For example, $H(\text{water}) = \left\{ 5^2 + \text{code(w)} \cdot \text{code(a)} + \text{code(a)} \cdot \text{code(t)} \right\} \bmod 9 = 7.$

The code for the letters is given in the table below:

letters	a,b,c,d,e	f,g,h,i,j,k	l,m,n,o,p,q	r,s,t,u,v	w,x,y,z
code(*letter*)	6	3	5	4	2

Assume the keys are: (1) apple; (2) banana; (3) orange; (4) mango; (5) kiwi; (6) potato.

- **a.** Compute $H(key)$ for each of the above keys.
- **b.** Insert the above keys (in exact same order) in a hash table with external chaining.
- **c.** Insert the keys in a hash table with open addressing (linear rehashing scheme).
- **d.** Compute the average number of probes for an unsuccessful search in the hash table of part (c).

(OVER)

4. [Marks 20]

For the following numbers:
39, 64, 41, 99, 53, 21, 83, 46, 7, 56.
Draw the diagrams to construct a max heap using: **(a)** siftdown (heapify); **(b)** siftup.

5. [Marks 10]

Construct a binary tree whose *preorder* traversal is: EADBC, while it's *inorder* traversal is: AEBCD. If no such tree is possible then you need to justify it.

6. [Marks 10+5=15]

Consider the Binary Tree (BT) ADT specifications, write the code of the function with the given prototype and specifications:

a. Write the client function:

```
template <class Type>
bool isLeaf(BT<Type> &t);
```

Precondition: BT is not empty.
Process: returns true if the current node is a leaf node and false otherwise.

b. Write the method (member function):

```
template <class Type>
bool BT<Type>::isLeaf();
```

Precondition: same as in part (a).
Process: same as in part (a).