KING SAUD UNIVERSITY

COLLEGE OF COMPUTER & INFORMATION SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

CSC212 Data Structures
Second Semester 1425/1426 AH

First Mid-term Examination:	Tuesday 26.02.1426 A.H./05.04.2005 C.E.
Duration:	2 Hours
Instructors:	Dr. Aqil Azmi and Mr. Saad Al-Ahmadi

---

1. [Marks 10+6+3+6=25]

a. Give a simplified Big-*O* bound for the following functions:

1. $10n^3 - 20n + 100$
2. $\log(n^2 + 1)$
3. $n(n! + 2^n)$
4. $3n^2 \big/ (n+1)$
5. $\sqrt{n \log 3^n}$

b. For each of the following pairs of functions $f(n)$ and $g(n)$, either $f(n) = O(g(n))$ or $g(n) = O(f(n))$, but not both. Determine which is the case,

1. $f(n) = n + \log n,\ \ g(n) = n\sqrt{n}.$
2. $f(n) = 2(\log n)^2,\ \ g(n) = 1 + \log n.$
3. $f(n) = 2^n,\ \ g(n) = 1000n^{10}.$

c. What is the Big-*O*, C, n_0 for the equation $(5n^2 + 10)(n^3 - 1)$

d. Consider the following code fragment. What value of x will it output?

```
x = 10;
for (k = 1; k <= 100; ++k)
   for (i = k; i <= 100; ++i)
      x = x + 2;
cout << "x=",x;
```

Show all your calculations.

***Hint*:** useful formula, $\sum_{j=i}^{n} j = \frac{1}{2}[n(n+1) - i(i-1)]$

2. [Marks 7+8=15]

Consider the Link list ADT specifications, write the code of the function with the given prototype and specifications:

a. Write the client function:

```
template <class Type>
void findprior(Linklist<Type> &list);
```

Precondition: (1) link list not empty; (2) current is not pointing at the first node.
Process: the node prior (before) the current node is made the current node.

b. Write the method (member function) using single pointer implementation:

```
template <class Type>
void Linklist<Type>::findprior();
```

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

3. [Marks 20]

Using the Sequential list ADT write the client function:

```
template <class Type>
void RemoveDuplicates(List<Type> &list);
```

Precondition: none.

Process: removes all the duplicate elements from the list. Leaving the first occurrence of the element, all subsequent occurrences are removed from the list. See the example below,

before	*3*	*5*	*9*	*3*	*7*	*10*	*5*	*3*	*6*	*7*
after	*3*	*5*	*9*	*7*	*10*	*6*				

4. [Marks 20]

Write the client function,

```
void PrintCommon(Stack<char> &SA, Stack<char> &SB);
```

which receives two stacks `SA` and `SB` (both of type `char`) and prints the common elements (العناصر المشتركة). That is, elements which appears in both stacks (case sensitive). For example if input was,

Stack SA	*a*	*b*	*c*	*d*		
Stack SB	*c*	*g*	*A*	*f*	*b*	*k*

then the code should output: *c*, *b*.

Note: (1) do not use any mass storage, *i.e.* no arrays, link lists, queues and extra stacks; and (2) upon return both stacks should be intact (المحافظة على محتوياتهما بعد انتهاء الدالة).

5. [Marks 20]

Assume that you have only the Queue ADT. Implement the complete Stack ADT using the Queue ADT only. You are not allowed to use arrays, link list, other stacks.

المطلوب: تنفيذ جميع عمليات Stack ADT باستعمال ال Queue ADT فقط.