Algorithm Design & Analysis

Assignment-3 (Basic Algorithms)

1. Suppose that each row of an $n \times n$ array *Arr* consists of 1's and 0's such that, in any row of *Arr*, all the 1's come before any 0's in that row. Assuming *Arr* is already in memory, write a program running in $O(n)$ time for finding the row of *Arr* that contains the most 1's.

2. You are given two linked lists A and B which may or may not contain a common node. From the first common node (if any) in A and B, the two lists are the same until the end. The two lists in presence of a common node look like the Roman letter Y.

Part (a)

Read the total number *n* of nodes in the lists A and B from the user. Create the two lists together as follows:

```
while (a total of n nodes are not added) {
    Create a new node with randomly generated data;
    If the two lists are already merged, append this new node to the merged part,
    and continue;
    Make a random decision whether the two lists are to be merged at this new node;
    If so, merge the two lists, and continue;
    Append the new node to one of the two lists (each with probability half);
}
```

Print the two lists individually.

Part (b)

Write a function to print the lists together. You pass only the two list headers as parameters to your function. The function first prints the elements of A before the first common node. It then prints the elements of B before the first common node. Finally, the nodes common to both the lists are printed. Your function must not make any change to the lists A and B.

Sample Output

n = 20

+++ Part(a)

List A : 832 494 318 17 23 569 389 298 488 19 144 849
List B : 225 262 188 637 353 605 300 589 23 569 389 298 488 19 144 849

+++ Part(b)

Initial part of A : 832 494 318 17
Initial part of B : 225 262 188 637 353 605 300 589
Common part : 23 569 389 298 488 19 144 849

3. You are given an $n \times n$ board. Your task is to place m coins on the board such that no two of the coins go to the same cell or to two adjacent cells. Two cells are called adjacent if their boundaries share an edge. Two cells with boundaries sharing only a corner will not be called adjacent. Write a program that prints all possible arrangements of m coins in the $n \times n$ board. Your program must not use any global or static variables.

Sample Output

For n= 4 and m= 7. The output is shown in a multi-column format to save space.

```
Arrangement 1:
x . x .
. x . x
x . x .
. x . .
Arrangement 2:
x . x .
. x . x
x . x .
. . . x
Arrangement 3:
x . x .
. x . x
x . . .
. x . x
Arrangement 4:
x . x .
. x . x
. . x .
x . . x
Arrangement 5:
x . x .
. x . x
. . x .
. x . x
Arrangement 6:
x . x .
. x . .
x . x .
. x . x
Arrangement 7:
x . x .
. . . x
x . x .
. x . x
Arrangement 8:
x . . x
. x . .
x . x .
. x . x
Arrangement 9:
x . . x
. . x .
. x . x
x . x .
Arrangement 10:
x . . .
. x . x
x . x .
. x . x
Arrangement 11:
. x . x
x . x .
. x . x
x . . .
Arrangement 12:
. x . x
x . x .
. x . x
. . x .
Arrangement 13:
. x . x
x . x .
. x . .
x . x .
Arrangement 14:
. x . x
x . x .
. x . .
x . . x
Arrangement 15:
. x . x
x . x .
. . . x
x . x .
Arrangement 16:
. x . x
x . . .
. x . x
x . x .
Arrangement 17:
. x . x
. . x .
. x . x
x . x .
Arrangement 18:
. x . .
x . x .
. x . x
x . x .
Arrangement 19:
. . x .
. x . x
x . x .
. x . x
Arrangement 20:
. . . x
x . x .
. x . x
x . x .
```

4. You are provided an unsorted array A having size n that may or may not contain duplicates and a number k (where, $k < n$). Your task is to write a program that runs in $O(n)$ time and returns true if array A contains duplicates within a distance of k, i.e. there exists $i, j \in [0, n-1]$ such that $A[i] = A[j]$ and $|i - j| < k$.