UCS 1312 Data Structures Lab
A1: Array ADT and its application

--Dr. R. Kanchana

Best Practices to be adapted

Design before coding
Modular design and coding using versions
Uniform notation for pseudo-code
Verification of algorithm using Hand-trace
Use of Multi-file C program

^^

Design the algorithm and implement in C.

1. Create an ADT for an array data structure with the following functions:
   a. *insertAt(A[ ], size, pos, data)* that inserts *data* at position *pos* in the array *A[size]* and returns size of the array if successful or -1 if not successful.
   b. *search(A[ ], pos, key)* that searches *key* in *A[size]* starting from pos and return the index of key if found or 0 if not found
   c. *size(A[ ])* that returns the length of the array *a*
2. Store arrayADT operations in Array.h
3. Use Array.h and write an application (main.c) for the following:
   a. Create a user interface that inserts a set of integers in array ADT. Do not take size of the array as input.
   b. Implement *insertafterdata(a[ ], data1, data2)* that inserts *data2* after every occurrence of *data1* in *a*.
   c. Write a function *printArray(a[ ])* that prints the integers in *a* with its position horizontally

Eg. Input:

a[7]	45	13	25	13	43	25	13			
data1	13									
data2	33									
Output										
a	1	2	3	4	5	6	7	8	9	10
	45	13	33	25	13	33	43	25	13	33

****