MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG213 Data Structures – Worksheet 5

Linked Lists Operations

Purpose:
This worksheet is concerned with the implementations of general operations for linked lists abstract data types. Also, you will implement some algorithms on linked lists to make you more comfortable with the concept. You are not expected to complete the entire worksheet in class. Work on as many problems as you can; the remaining problems you can use for practice and to test your understanding of the application of linked lists.

You may use the following declaration for linked lists:

```
struct Node
{
  int val;
  struct Node *next;
};

struct ListRecord
{
  struct Node *head;
  struct Node *tail;
  int size;
};
typedef struct ListRecord *List;
```

For the questions from 1 to 6, you have to use the given CNG213_Worksheet5_Template.c file

In the template program given to you some basic functions are already implemented which includes: `CreateList()`, `MakeEmptyList()`, `ListSize()`, `HeadOfList()`, `TailOfList()`, `IsEmptyList ()`, `DisplayList()`. By using the given template implement the following functions. Make sure that you update the corresponding main function to test these functions.

1) Write a function called `InsertList()` which given a list, a position and a value creates a node to store the given value and inserts this node to the given position.
2) Write a function called `DeleteList()` which given a value finds the node in the list with that value and deletes that node from the list.
3) Write a function called `GetElementAtPosition()` which given a list and a position value, finds the node at that position and returns the value stored in the node in that position.
4) Write a function called `GetPositionOfElement()` which given a list and a value, finds the position of the node that stores that given value.
5) Write a `SortedInsert()` function which given a list that is sorted in increasing order, and a single node, inserts the node into the correct sorted position in the list. Note that this function does not allocate any memory space; it simply takes an existing node, and just rearranges pointers to insert it into the list. There are many possible solutions to this problem.
6) Write a C function `InsertSort()` that given a list, rearranges the elements in the list in ascending order. It should use `SortedInsert()`.

Please note that from this worksheet, you need to submit the solution of the following question (7)

7) Write a program which will do following;

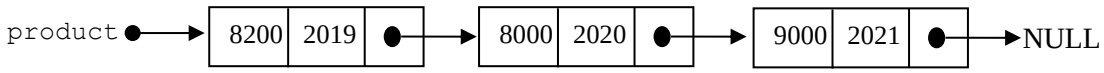
a) `Node` declaration of a Linked List with year and prices of a product and `ListRecord` declaration.

b) In the function `main()` create a linked list of the `product` and fill it with the year and prices taken from the user.

c) A function `PrintProducts()` that takes the linked list of product as an input, traverses and prints the prices.

d) A function `findCheapest()` that takes the linked list of product as an input, traverses and returns the cheapest product. Display should take place in `main()`.

If the input prices are as 8200, 8000 and 9000 and years are 2019, 2020 ,2021 (please note that you cannot make any assumptions regarding to the size of the list (number of years and products)), then list formed by `main()` should be as follows;



A sample run would be as follows:

```
**************************************************
1) Create yearly price for the product
2) Display yearly price for the product
3) Display the cheapest product info
4) Exit
What would you like to do? 1
**************************************************
How many years? 3
Enter price and year for the product: 8200 2019
Enter price and year for the product: 8000 2020
Enter price and year for the product: 9000 2021
Prices and year info for the product is created successfully!
**************************************************
1) Create yearly price for the product
2) Display yearly price for the product
3) Display the cheapest product info
4) Exit
What would you like to do? 2
**************************************************
Yearly price for the product
2019: 8200 £
2020: 8000 £
2021: 9000 £
**************************************************
1) Create yearly price for the product
2) Display yearly price for the product
3) Display the cheapest product info
4) Exit
What would you like to do? 3
**************************************************
The cheapest price of the product is 8000£ in 2020!
**************************************************
1) Create yearly price for the product
2) Display yearly price for the product
3) Display the cheapest product info
4) Exit
What would you like to do? 4
```