

Lab 01: Review

1 Pointer

Use the Pointer technique to complete the following functions:

1. Input a n-element integer array with `int *a` is the pointer point to the allocated dynamic memory:

- **void InputArray(int* &a, int &n);**

2. Remove allocated dynamic memory:

- **void DellocateArray(int* &a);**

3. Output all elements of the array:

- **void PrintArray(int* a, int n);**

4. Find the smallest value from the array:

- **int FindMin(int* a, int n);**

5. Find the greatest absolute value from the array:

- **int FindMaxModulus(int* a, int n);**

6. Check if the array is ascending:

- **bool IsAscending(int* a, int n);**

7. Find the total value of all elements of the array:

- **int SumOfArray(int* a, int n);**

8. Count the number of prime numbers in the array:

- **int CountPrime(int* a, int n);**

9. Create a new dynamic array which is the reverse of the given array:

- **int ReverseArray(int* &a, int* b, int n);**

From Ex 10. to Ex 13. are Searching Algorithms. Return the first position found, else, return `-1`.

10. Sequential Search:

- **int LinearSearch(int* a, int n, int key);**

11. Sequential Search (using flag):

- **int SentinelLinearSearch(int* a, int n, int key);**

12. Binary Search:

- **int BinarySearch(int* a, int n, int key);**

13. Binary Search (using recursion):

- **int RecursiveBinarySearch(int* a, int left, int right, int key);**

2 Recursion

Use the Recursion technique to complete the following functions:

1. Find the total value of all integers that less than or equal to n : $S = 1^2 + 2^2 + \dots + n^2$.
 - **int SumOfSquares(int n);**
2. Find the greatest common divisor of 2 integers **a** and **b**:
 - **int GCD(int a, int b);**
3. Find the n^{th} Fibonacci number using by the following formular: $F(n) = F(n - 1) + F(n - 2)$.
 - **int FIB(int n);**

3 Linkedlist

Given the following Linkedlist definition:

```
struct NODE{
    int key;
    NODE* p_next;
};
```

```
struct List{
    NODE* p_head;
    NODE* p_tail;
};
```

Write the function to fulfill the following requirements:

1. Initialize a **NODE** from a given integer:
 - **NODE* CreateNode(int data);**
2. Initialize a **List** from a give **NODE**:
 - **List* CreateList(NODE* p_node);**
3. Insert an integer to the head of a given **List**:
 - **bool AddHead(List* L, int data);**
4. Insert an integer to the tail of a given **List**:
 - **bool AddTail(List* L, int data);**
5. Remove the first **NODE** of a given **List**:
 - **void RemoveHead(List* L);**
6. Remove the last **NODE** of a given **List**:
 - **void RemoveTail(List* L);**
7. Remove all **NODE** from a given **List**:
 - **void RemoveAll(List* &L);**
8. Print all elements of a given **List**:
 - **void PrintList(List* L);**
9. Count the number of elements **List**:
 - **int CountElements(List* L);**
10. Create a new **List** by reverse a given **List**:
 - **List* ReverseList(List* L);**
11. Remove all duplicates from a given **List**:
 - **void RemoveDuplicate(List* &L);**
12. Remove all **key** value from a given **List**:
 - **bool RemoveElement(List* &L, int key);**

4 Submission regulation

- Only files with **.cpp** and **.h** extension is required.
- Each part of this Lab must have its own folder.
- All folders must be put in the folder named as student's ID.
- Compress the folder into Student's ID.rar(.zip) for submission.
- Submission with wrong regulation will result in a "0" (zero).
- Plagiarism and Cheating will result in a "0" (zero) for the entire course and will be subject to appropriate referral to the Management Board of APCS for further action.