# Assignment #1

## CS 2001 – Data Structures (CS)
## Fall 2024

---

## General Guidelines

1. Peer plagiarism and the late submissions are strictly not allowed
2. Total Marks: 100

## Submission Guidelines

3. Your assignment submission will be on CLASSROOM within the givendeadline.
4. Analytical and mathematical questions can be handwritten, while code questions should preferably be computer-typed
5. **Deadline (14-09-2024)**

**Task 1:** Determine the dominating terms from the given running time functions and express them inform of Biḡ-O notation**.**

**A)**

```
i = 1;
while (i < n) {
            tot += i;
            i = i * 2;
}
```

**B)**

```
i = n;
while (i > 0) {
            tot += i;
            i = i / 2;
}
```

**C)**

```
for ( i=0 ; i<n ; i++ )

      for( j=0 ; j<n ; j++ )
                   sum[i] += entry[i][j][0];
```

**D)**

```
for ( i=0 ; i<n ; i++ )
   for( j=0 ; j<n ; j++ )
        for( k=0 ; k<n ; k++ )
            sum[i][j] += entry[i][j][k];
```

**E)**

```
      for ( i=0 ; i<n ; i++ )
              for( k=0 ; k<n ; k++ )
                   sum[i] += entry[i][0][k];
```

**F)**

```
for ( i=0 ; i<n ; i++ )
        for( j=0 ; j< sqrt(n) ; j++ )
            m += j;
```

**G)**

```
for ( i=0 ; i<n ; i++ )
{
     m += j;
     m += j;
     m += j;
     …
     m += j;   // 31 times
}
```

H)

```
 int sum = 0;
 for (int i = 1; i
 =n; i++) {
  for (int j = 1; j
 <= i; j++) {
sum += 1;
}
 }
```

***I)***

```
int sum = 0;
for (int i = 0; i < n; ++i) {
    for (int j = i; j <= 2*i; ++j) { sum
      += 1;
    }
  }
```

**J)**

```
int sum = 0;
for (int i =
1; i <= n;
i*= 2) {
    for (int j = 0; j < i; j++) {
      sum += 1;
    }
}
```

**Task 2:** Write a C++ program that will implement the CRUD operations for a singly LinkedList, additionally add comments to each function indicating the **worst-case complexity**. For your ease you can consider the LinkedList ADT given bellow:

```
LinkedList {
private:
        Node* head;
public:
        void insertNodeAtBeginning(int data);
        void insertNodeInMiddle(int data, int key);
        //will search for key and insert node after the node
        //where a node's data==key
        void insertNodeAtEnd(int data);
        bool deleteFirstNode();
        bool deleteNode(int key);
        bool deleteLastNode();
        void display();
        bool search(int data);
};
```

**Sample ADT of a Linked List**

**Task 3:** Write a C++ program that will implement the CRUD operations for **doubly** and **circular** LinkedList, additionally add comments to each function indicating the worst-case complexity. For yourease you can consider the LinkedList ADT given bellow:

```
LinkedList {
void insertNodeAtBeginning(int data);
void insertNodeInMiddle(int data, int key); //will search for
key and insert node after the node where a node's data==key
void insertNodeAtEnd(int data);
bool deleteFirstNode();
bool deleteNode(int key);
bool deleteLastNode();
void display();
bool search(int data);
};
```

**Sample ADT of a LinkedList**

Make sure to implement the above ADT for both Doubly LinkedList and Circular LinkedList to obtainfull marks.

**Task 4:** Your task is to use the singly linked list design in Task 02, then reverse the linked list by changing the links between nodes.

**Task 5:** Your task is to use the doubly linked list used in task 03 and sort that unsorted linked list.

**Task 6:** Given a singly linked list in task 1. The task is to remove duplicates (nodes with duplicate values)from the given list (if it exists).

**Note:** Try not to use extra space. The nodes are arranged in a **sorted Way**

**Task 7:** Write a program that prompts the user to input a string and then outputs the string in the pigLatin form. Input the string in a doubly circular link-list.

The rules for converting a string into pig Latin form are as follows:

If the string begins with a vowel, add the string "-way" at the end of the string. For example, the pig Latin form of the string "eye" is "eye-way". If the string does not begin with a vowel, first add "-" at the end of the string. Then rotate the string one character at a time; that is, move the first character of the string to the end of the string until the first character of the string becomes a vowel. Then add the string "ay" at the end. For example, the pig Latin form of the string "There" is "ere-Thay Strings such as "by" contain no vowels.

In cases like this, the letter y can be considered a vowel. So, for this program the vowels are a, e, i, o, u, y, A, E, I, O, U and Y. Therefore, the pig Latin form of "by" is "y-bay". Strings such as "1234" contain no vowels. The pig Latin form of the string "1234" is"1234- way". That is, the pig Latin form of a string that has no vowels in it is the string followed by the string "-way".

**Note:** Your program must store the characters of a string into a linked list and use the function rotate, to rotate the string.

## Best of Luck ◆

# Keep Exploring