



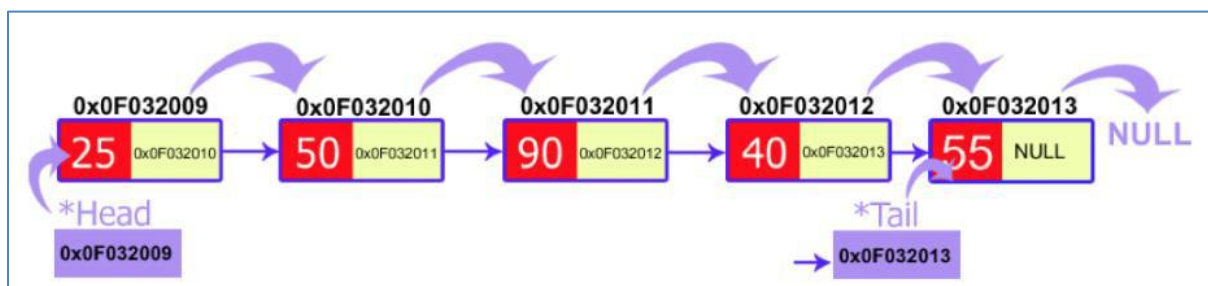
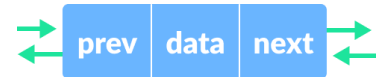
វិទ្យាស្ថានបច្ចេកវិទ្យាកម្ពុជា
Institute of Technology of Cambodia

TP-15
Linked List
in C++

Academic Year: 2022- 2023

1. What is Linked list?

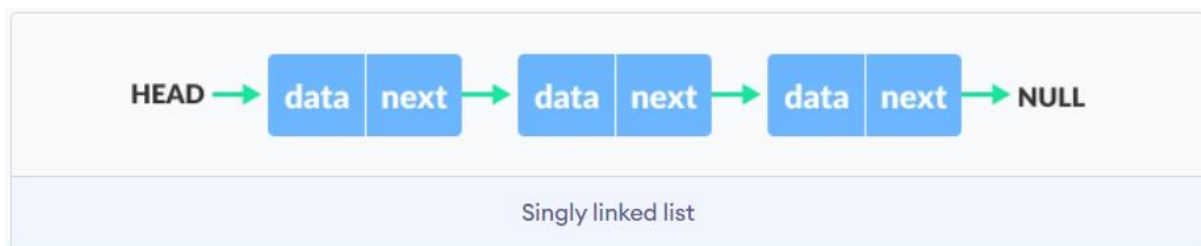
- A linked list is a data structure that can store an indefinite amount of elements (dynamic size).
- Each element contains
 - Data
 - A link (pointer)
 - to its **next** element (**successor**)
 - and/or to its **previous** element (**predecessor**)
- In a linked list, each element is linked with each other.
 - Element = called a **node**
 - the first element is **head** and the last element is **tail**



2. Type of Linked List

There are two types of linked lists:

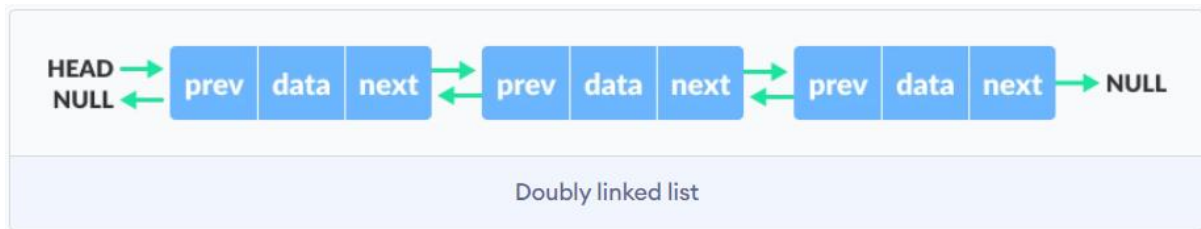
- A **singly linked list** is a linked list that has a link to either its **successor**.



Node is represented as:

```
struct node {
    int data;
    struct node *next;
}
```

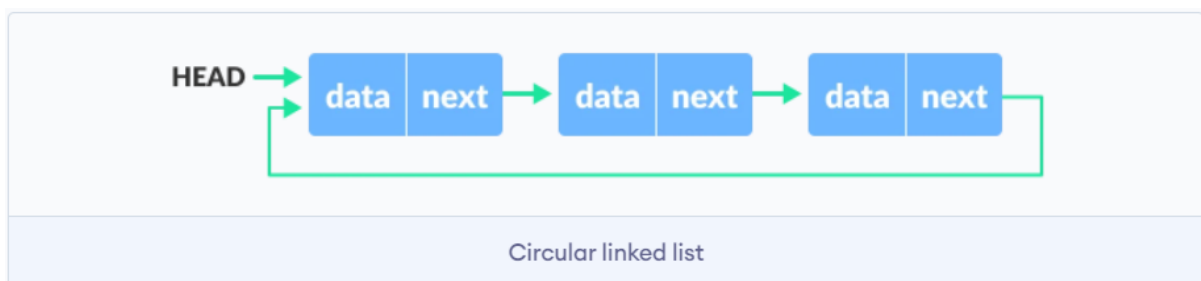
- A **doubly linked list** is a linked list that has **both links** to **successor** and **predecessor**.



Node is represented as:

```
struct node {
    int data;
    struct node *next;
    struct node *prev;
}
```

- A **singly** or **doubly linked list** can be called a **circular linked list** when the last element (**tail**) points to the first element (**head**).



- for **singly linked list**, **next** pointer of last item points to the first item
- In the **doubly linked list**, **prev** pointer of the first item points to the last item as well.

3. Array Vs. Linked List

Difference between **Array** and **Linked List**:

	Array	Linked List
Strength	<ul style="list-style-type: none"> • Random Access (Fast Search Time) • Less memory needed per element • Better cache locality 	<ul style="list-style-type: none"> • Fast Insertion/Deletion Time • Dynamic Size • Efficient memory allocation/utilization
Weakness	<ul style="list-style-type: none"> • Slow Insertion/Deletion Time • Fixed Size • Inefficient memory allocation/utilization 	<ul style="list-style-type: none"> • Slow Search Time • More memory needed per node as additional storage required for pointers

Write C++ Program to solve the problem below:

1. We want to store a list of all engineer students' ID at ITC (more students are added every year thus the list should be able to store unlimited number of student IDs). Define a linked list data structure for this kind of problem. Hint: Create a structure element and list. Data in element is ITC's student ID.

2. In addition to problem #1, create a function to create an empty list, a function to add data to the list, a function to display all data in the list. In main program, call these three functions to test and see the result.

3. Create a singly linked list that can store integer numbers. Create 4 functions i) create list, ii) add data to end of list, iii) add data to beginning of list, iv) display data in list. Then ...

- Create an empty list
- Add the number 7 to the beginning of the list
- Add 1 to the beginning of the list
- Display all numbers in the list
- Add 0 to the end of the list
- Add the number 4 to the end of the list
- Display all numbers in the list

4. Create a singly linked list for storing information of students. Each student has id, name, and average score. Then write a program to

- Create a function to ask for information (id, name, average score) for a student then add her/him to the list. The program does not add the student to the list when the input id is already exists in the list.
- Create a function to display information of all students in the list.

5. Get many numbers from a user and store in a singly linked list. The program asks a user for a number then store in the list. When users input 0, stop asking user for more numbers. Display all data in the list. Find summation of all data in the list and show the result.

Reference Pyramid Pattern:

<https://www.programiz.com/dsa/linked-list>

<https://www.programiz.com/dsa/linked-list-operations>

<https://www.geeksforgeeks.org/linked-list-vs-array/>