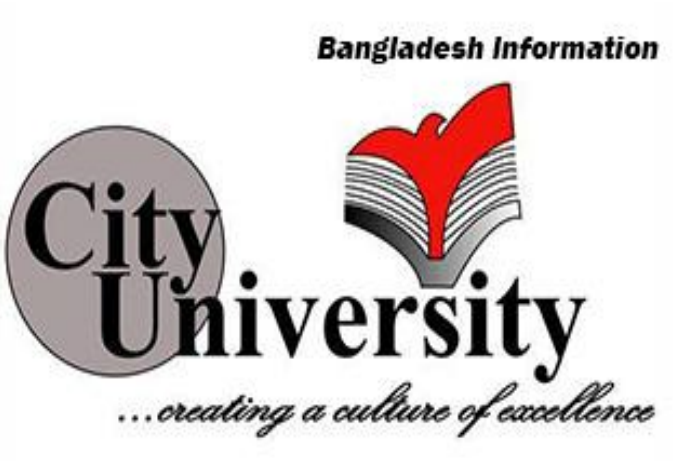


Lab Report

Course Title: Data Structure Laboratory

Course Code: CSE 214

Submitted To:

Richard Philip

Lecturer

Department of CSE

City University

Submitted By:

Dolon Akter

ID: 171442561

Batch: 44th

Program: CSE

Data Structure:

Mid Term:

1. Array,
2. Stack,
3. Queue,
4. Linked list.

Final Term:

Index

Mid Term

No	Topic	Page
1.	Array	
2.	Stack	
3.	Queue	
4.	Linked List	

Section:

1. **Array – 1D Array,**
   2D Array.
2. **Stack – Stack push using array,**
   Stack pop using array.
3. **Queue – Queue using array.**
4. **Linked List – Single linked list,**
   Double linked list.

- **Array:**

  An **array** is stored such that the position of each element can be computed from its index tuple by a mathematical formula. The simplest type of **data structure** is a linear **array**, also called one dimensional **array**.

  1D Array:

```
#include<stdio.h>

int main()
{
int a[10];
int i, n;

printf("How many element you want to save \n");
scanf("%d", &n);

printf("enter element one by one \n");


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

scanf("%d", &a[i]);

}

printf("The List you entered\n");

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

```
printf("%d ", a[i]);

}
return 1;
}
```

2D Array:

The 2D array is organized as matrices which can be represented as the collection of rows and columns. int A[rows][columns];

- Wikipedia

```
#include<stdio.h>

void main(){

int a[10][10];

int i,j,rows, columns;
printf("How many rows you want \n");
scanf("%d", &rows);
printf("How many columns you want \n");
scanf("%d", &columns);

printf("Enter your array Element one by one
\n");

for(i=0;i<rows;i++){

    for(j=0;j<columns;j++){

          scanf("%d", &a[i][j]);
    }
```

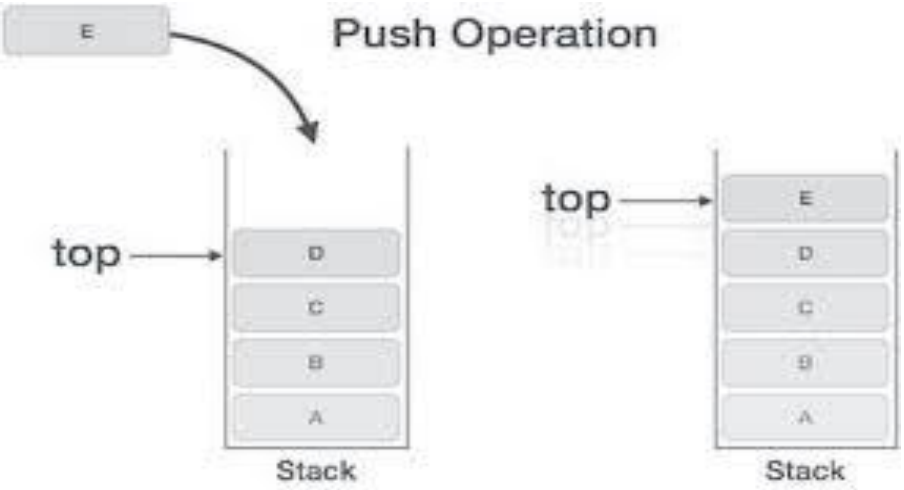
```
}
printf("your array \n");

for(i=0;i<rows;i++){

    for(j=0;j<columns;j++){

          printf("%d ", a[i][j]);
    }
    printf("\n");
}


}
```

- Data_Structure_Notes.pdf

Stack:

Stack is an ordered list of similar **data** type. **Stack** is a LIFO (Last in First out) **structure** or we can say FILO (First in Last out). Push () function is used to insert new elements into the **Stack** and pop () function is used to remove an element from the **stack**.

- Studytonight.com



- Tutorialspoint.com

Stack Push Using Array:

```
void push( int data) {
top++;
stack [ top ] = data ;
}
```

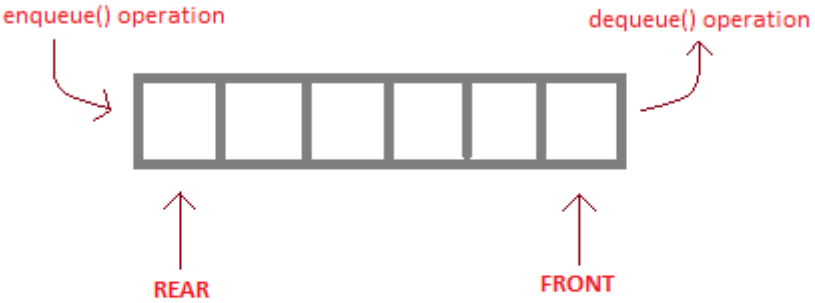
Stack Pop Using Array:

```
int pop() {
int data ;
data = stack [ top ] ;
top−−;
return data ;
}
```

- **Data_Structure_Notes.pdf**

- **Queue:**

  Queue is an abstract **data structure**, somewhat similar to Stacks. Unlike stacks, a **queue** is open at both its ends. One end is always used to insert **data** (enqueue) and the other is used to remove **data** (dequeue). **Queue** follows First-In-First-Out methodology, i.e., the **data** item stored first will be accessed first.



enqueue( ) is the operation for adding an element into Queue.
dequeue( ) is the operation for removing an element from Queue .

QUEUE DATA STRUCTURE

- **Tutorialspoint.com**

Queue using array:

```
void enqueue( int data) {
head++;
stack [ head ] = data ;
 }
```

```
int dequeue () {
 int data ;
data = stack [ tail ] ;
tail++;
return data ;
}
```

- Data_Structure_Notes.pdf

Linked List:

In computer science, a **linked list** is a linear collection of **data** elements, whose order is not given by their physical placement in memory. Instead, each element points to the next. It is a **data structure** consisting of a collection of nodes which together represent a sequence.

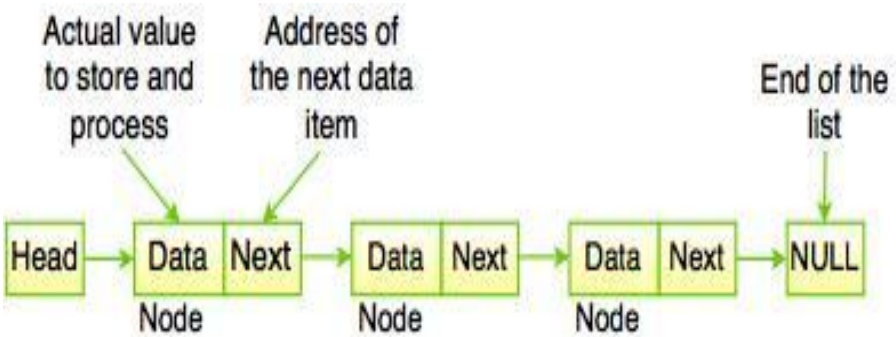


Fig. Linked List

- Wikipedia

Single Linked List:

Singly Linked Lists are a type of data structure. In a **singly linked list**, each node in the **list** stores the contents and a pointer or reference to the next node in the **list**. It does not store any pointer or reference to the previous node. ... The last node in a **single linked list** points to nothing.

- Wikibooks.com

```
Struct Node
{
Int data;
Struct Node*next;
};
```

Double Linked List:

In computer science, a **doubly linked list** is a linked data structure that consists of a **set** of sequentially linked records called nodes. Each node contains two fields, called links, that are references to the previous and to the next node in the sequence of nodes.

```
Struct Node
{
```

Int data;

Struct Node*next;

Struct Node*prev;

};

- Wikipedia