

25/1/24

## 2. Stack implementation using linked list

```
#include <stdio.h>
void push();
void pop();
void display();
struct node
{
    int data;
    struct node * next;
}
struct node * head = NULL;

void main()
{
    int ch;
    printf("Enter : 1. Push 2. Pop 3. Display:");
    scanf("%d", &ch);
    while (ch != 4)
    {
        switch (ch)
        {
            case 1 : push();
                     break;
            case 2 : pop();
                     break;
            case 3 : display();
                     break;
        }
    }
    printf("Exited");
}
```

Exited  
25/1/24

```
void push()
```

```
{
```

```
int data, n;
```

```
printf("Enter no. of nodes:");
```

```
scanf("%d", &n);
```

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

```
{
```

```
struct node *last = head;
```

```
struct node *new-node
```

```
= (struct node*) malloc  
(sizeof(struct node));
```

```
printf("Enter the data:");
```

```
scanf("%d", &new-node->data);
```

```
new-node->next = NULL;
```

```
while (last->next != NULL)
```

```
last = last->next;
```

```
last->next = new-node;
```

```
}
```

```
}
```

```
void pop()
```

```
{
```

```
struct node *ptr;
```

```
struct node *ptr1;
```

```
if (head->next == NULL)
```

```
{ free(head);
```

```
head = NULL;
```

```
printf("Element popped");
```

```
}
```

```
else
```

```
{
```

```
ptr = head;
```

```

ptr1 = head;
while (ptr -> next != NULL)
{
    ptr1 = ptr;
    ptr = ptr -> next;
}
ptr1 -> next = NULL;
free(ptr);
printf("Element popped successfully");
}
}

```

```

void display()
{
    struct node *p = head;
    printf("Stack: \n");
    while (p != NULL)
    {
        printf("%d -> ", p->data);
        p = p->next;
    }
    printf("NULL \n");
}

```



## OUTPUT :

1. Push

2. Pop

3. Display

4. Exit

Enter choice : 1

Enter the data : 5

Element pushed successfully

Enter choice : 1

Enter the data : 4

Element pushed successfully

Enter choice : 1

Enter the data : 3

Element pushed successfully

Enter choice : 3

List:

5 → 4 → 3 → NULL

Enter choice : 2

Element popped successfully

Enter choice : 3

List

5 → 4 → NULL

Enter choice : 4

Exited

25/1/24

Bafna Gold

Date: Page:

## 3. Queue Implementation using linked list

#include &lt;stdio.h&gt;

void enqueue();

void dequeue();

struct node

{

int data;

struct node \*next;

};

struct node \*head = NULL;

void main()

{

printf("1. Enqueue 2. Dequeue 3. Display 4. Exit");

int ch;

while(ch != 4)

{

printf("Enter choice: ");

scanf("%d", &amp;ch);

switch(ch)

{

case 1: enqueue();

break;

case 2: dequeue();

break;

case 3:

display();

break;

}

}

}

```
void enqueue()
```

```
{
```

```
    int data;
```

```
    struct node *last = head;
```

```
    struct node *new-node;
```

```
    new-node = (struct node*) malloc  
                (sizeof(struct node));
```

```
    printf("Enter the data:");
```

```
    scanf("%d", &data);
```

```
    new-node->data = data;
```

```
    new-node->next = NULL;
```

```
    if(head == NULL)
```

```
        head = new-node;
```

```
    else
```

```
    { while(last->next != NULL)
```

```
        last = last->next;
```

```
        last->next = new-node;
```

```
    } printf("Node added successfully\n");
```

```
}
```

```
void dequeue()
```

```
{
```

```
    struct node *ptr;
```

```
    if(head == NULL)
```

```
        printf("List is empty\n");
```

```
    else
```

```
    { ptr = head;
```

```
        head = ptr->next;
```

```
        free(ptr);
```

```
    } printf("Node deleted from beginning\n");
```

```
}
```



```
void display()
{
    struct node *p = head;
    printf("\n");
    while (p != NULL)
    {
        printf("%d -> ", p->data);
        p = p->next;
    }
    printf("NULL\n");
}
```

OUTPUT :

Enter :

1. Enqueue

2. Dequeue

3. Display

Enter choice : 1

Enter data : 1

Enter choice : 1

Enter data : 2

Enter choice : 1

Enter data : 3

Enter choice : 3

~~list~~ Queue:

1 → 2 → 3 → NULL

Enter choice : 2

Node deleted from beginning

Enter choice : 3

Queue:

2 → 3 → NULL

Enter choice : 4

Exited