

2.a. Stack implementation using SLL

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct node {
```

```
    int data;
```

```
    struct node * next;
```

```
};
```

```
struct node * top = NULL;
```

```
void push()
```

```
{
```

```
    struct node * new_node;
```

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

```
    printf("Enter the element\n");
```

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

```
    new_node->next = NULL;
```

```
    if (top == NULL) {
```

```
        top = new_node;
```

```
    } else {
```

```
        new_node->next = top;
```

```
        top = new_node;
```

```
    }
```

```
}
```

```
void pop()
```

```
{
```

```
    if (top == NULL) {
```

```
        printf("Stack is empty");
```

```
    } else {
```

```
        printf("deleted item is %d", top->data);
```

```
        top = top->next;
```

```
    }
```

```
void display()
```

```
{
```

```
    struct node * temp;
```

```
    if (top == NULL) {
```

```
printf("Stack is empty");
```

```
} else {
```

```
printf("deleted item
```

```
temp = top;
```

```
while (temp != NULL) {
```

```
printf("%d\n", temp->data);
```

```
temp = temp->next;
```

```
}
```

```
}
```

```
}
```

```
int main () {
```

```
int choice;
```

```
while(1) {
```

```
printf("1. push\n2. pop\n3. display\n4. exit\nenter  
your choice: ");
```

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

```
switch (choice) {
```

```
case 1: push(); break;
```

```
case 2: pop(); break;
```

```
case 3: display(); break;
```

```
case 4: exit(0);
```

```
}
```

```
return 0;
```

```
}
```


% Stack

1. push
2. pop
3. display
4. exit

enter your choice : 1

enter your element
2

1. push
2. pop
3. display
4. exit

enter your choice : 2

deleted item is 2.

C:\Users\User\Desktop\1BM22CS312\stackll.exe

1.PUSH 2.POP 3.DISPLAY 4.EXIT

1

Enter a value11

Successfully Added 11

1.PUSH 2.POP 3.DISPLAY 4.EXIT

1

Enter a value11

Successfully Added 11

1.PUSH 2.POP 3.DISPLAY 4.EXIT

3

11

11

1.PUSH 2.POP 3.DISPLAY 4.EXIT

2.b. Queue Implementation

```
#include <stdio.h>
#include <stdlib.h>

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

struct node * front = NULL;
struct node * rear = NULL;

void enqueue() {
    struct node * new_node = (struct node *) malloc(sizeof(struct node));
    printf("Enter the element: ");
    scanf("%d", &new_node->data);
    new_node->next = NULL;
    if (rear == NULL) {
        front = rear = new_node;
    } else {
        rear->next = new_node;
        rear = new_node;
    }
}

void dequeue() {
    if (front == NULL) {
        printf("Queue is Empty\n");
    } else {
        struct node * temp = front;
        printf("Deleted item is %d\n", temp->data);
        front = front->next;
        free(temp);
    }
}
```

```
void display() {
```

```
    struct node * temp = front;
```

```
    if (temp == NULL) {
```

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

```
    } else {
```

```
        printf("Queue Elements: \n");
```

```
        while (temp != NULL) {
```

```
            printf("%d \n", temp->data);
```

```
            temp = temp->next;
```

```
        }
```

```
    }
```

```
int main() {
```

```
    int choice;
```

```
    while(1) {
```

```
        printf("1. Enqueue \n 2. Dequeue \n 3. Display \n 4. Exit \n\n Enter your choice: ");
```

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

```
        switch(choice) {
```

```
            case 1: enqueue(); break;
```

```
            case 2: dequeue(); break;
```

```
            case 3: display(); break;
```

```
            case 4: exit(0);
```

```
        }
```

```
    } return 0;
```

```
}
```

29/11/24

Enqueue
Dequeue

Stack, Queue

Sort

- Op:
1. Enqueue
 2. Dequeue
 3. Display
 4. Exit

Enter your choice : 1

Enter the element : 4

1. Enqueue
2. Dequeue
3. Display
4. Exit

Enter your choice : 6

 C:\Users\User\Desktop\1BM22CS312\qll.exe

1.Enqueue 2.Dequeue 3.DISPLAY 4.EXIT

1

Enter a value: 6

Successfully Added 6

1.Enqueue 2.Dequeue 3.DISPLAY 4.EXIT

1

Enter a value: 2

1.Enqueue 2.Dequeue 3.DISPLAY 4.EXIT

3

6

2

1.Enqueue 2.Dequeue 3.DISPLAY 4.EXIT