1.Give a program for stack using array

Program:

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

#define MAX 100
// Stack structure definition
struct Stack {
   int data[MAX];
   int top;
};
void initialize(struct Stack *stack)
 {
   stack->top = -1;
}
int isEmpty(struct Stack *stack)
{
   return stack->top == -1;
}
int isFull(struct Stack *stack)
 {
   return stack->top == MAX - 1;
}
void push(struct Stack *stack, int value)
{
   if (isFull(stack))
 {
      printf("Stack overflow! Cannot push %d\n", value);
      return;
   }
   stack->data[++(stack->top)] = value;
```

```
    printf("%d pushed to stack\n", value);
}
int pop(struct Stack *stack)
{
    if (isEmpty(stack))
{
        printf("Stack underflow! Cannot pop element\n");
        return -1;
    }
    return stack->data[(stack->top)--];
}
void display(struct Stack *stack)
{
    if (isEmpty(stack))
{
        printf("Stack is empty!\n");
        return;
    }
    printf("Stack elements are: ");
    for (int i = 0; i <= stack->top; i++)
{
        printf("%d ", stack->data[i]);
    }
    printf("\n");
}
int main()
{
    struct Stack stack;
    initialize(&stack);

    int choice, value;
```

```
  while(1)
{
    printf("\n1. Push\n2. Pop\n3. Display\n4. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch(choice)
{
        case 1:
            printf("Enter the value to push: ");
            scanf("%d", &value);
            push(&stack, value);
            break;
        case 2:
            value = pop(&stack);
            if(value != -1)
                printf("Popped value: %d\n", value);
            break;
        case 3:
            display(&stack);
            break;
        case 4:
            exit(0);
        default:
            printf("Invalid choice! Please try again.\n");
    }
  }

  return 0;
}
```

Output: 1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 1

Enter the value to push: 10

10 pushed to stack

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 1

Enter the value to push: 20

20 pushed to stack

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 1

Enter the value to push: 30

30 pushed to stack

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 3

Stack elements are: 10 20 30

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 2

Popped value: 30

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 2

Popped value: 20

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 3

Stack elements are: 10

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 4

2.Give a program for stack using linked list

Program:

```
#include <stdio.h>
#include <stdlib.h>
struct Node
{
   int data;
   struct Node* next;
};
struct Node* createNode(int value)
{
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = value;
   newNode->next = NULL;
   return newNode;
}
int isEmpty(struct Node* top)
{
   return top == NULL;
}
void push(struct Node** top, int value)
{
   struct Node* newNode = createNode(value);
   newNode->next = *top;
   *top = newNode;
   printf("%d pushed to stack\n", value);
}
int pop(struct Node** top)
{
   if (isEmpty(*top))
```

```
{
        printf("Stack underflow! Cannot pop element\n");
        return -1;
    }
    struct Node* temp = *top;
    int poppedValue = temp->data;
    *top = (*top)->next;
    free(temp);
    return poppedValue;
}
void display(struct Node* top)
{
    if (isEmpty(top)) {
        printf("Stack is empty!\n");
        return;
    }
    struct Node* temp = top;
    printf("Stack elements are: ");
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}
int main()
 {
    struct Node* stack = NULL;

    int choice, value;
    while(1)
 {
```

```
        printf("\n1. Push\n2. Pop\n3. Display\n4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch(choice) {
            case 1:
                printf("Enter the value to push: ");
                scanf("%d", &value);
                push(&stack, value);
                break;
            case 2:
                value = pop(&stack);
                if(value != -1)
                    printf("Popped value: %d\n", value);
                break;
            case 3:
                display(stack);
                break;
            case 4:
                exit(0);
            default:
                printf("Invalid choice! Please try again.\n");
        }
    }
    return 0;
}
```

Output:

1. Push
2. Pop
3. Display
4. Exit

Enter your choice: 1

Enter the value to push: 10

10 pushed to stack

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 1

Enter the value to push: 20

20 pushed to stack

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 1

Enter the value to push: 30

30 pushed to stack

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 3

Stack elements are: 30 20 10

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 2

Popped value: 30

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 2

Popped value: 20

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 3

Stack elements are: 10

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 4