1. Write a C program to implement the stack using switch case that includes operations like: Push, Pop, Peek and Display

Code:

```c
#include <stdio.h>
#include <stdlib.h>
#define max 5
int top = -1;
int stack[max];

void push() {
    if (top == max-1){
        printf("Error: Stack Overflow\n");
    } else {
        int val;
        printf("Enter the value: ");
        scanf("%d", &val);
        top++;
        stack[top] = val;
    }
}
void pop(){
    if (top == -1) {
        printf("Error: Stack Underflow\n");
    } else {
        printf("The element deleted: %d\n", stack[top]);
        top--;
    }
}
void peek() {
    if (top==-1){
        printf("Error: Stack Underflow\n");
    }
    else {
        printf("The current top value: %d\n", stack[top]);
    }
}
void display() {
    if (top == -1) {
        printf("Error: Stack Underflow\n");
    } else {
        for (int i = top; i >= 0; i--){
            printf("%d ", stack[i]);
        }
        printf("\n");
    }
```

```c
}

int main()
{
    int choice;
    int con=1;


    while(con == 1){
    printf("\n 1.Push\n 2.Pop\n 3.Peak\n 4.Display\n");
    printf("Choose a operation: ");
    scanf("%d", &choice);
    printf("\n");
        switch (choice) {
            case 1:
                push();
                break;
            case 2:
                pop();
                break;
            case 3:
                peek();
                break;
            case 4:
                display();
                break;
            default:
                printf("Invalid Operation\n");
                break;
        }
        printf("\nEnter 1 to continue: ");
        scanf("%d", &con);
        printf("\n");
    }
}
```

Output:

```
 1.Push
 2.Pop
 3.Peak
 4.Display
Choose a operation: 1
Enter the value: 2

Enter 1 to continue: 1

 1.Push
 2.Pop
 3.Peak
 4.Display
Choose a operation: 1
Enter the value: 3

Enter 1 to continue: 1

 1.Push
 2.Pop
 3.Peak
 4.Display
Choose a operation: 4
3 2

Enter 1 to continue: 1

 1.Push
 2.Pop
 3.Peak
 4.Display
Choose a operation: 2

The element deleted: 3

Enter 1 to continue: 1

 1.Push
 2.Pop
 3.Peak
 4.Display
Choose a operation: 3

The current top value: 2
```

2. Write a C program to implement the queue using switch case that includes operations like Enqueue, Dequeue, Display

Code:

```c
#include <stdio.h>
#include <stdlib.h>
#define max 5

int front = -1, rear = -1;
int queue[max];

void enqueue() {
  if (rear==max-1){
    printf("Queue Overflow\n");
    return;
  } else if (front == -1 && rear == -1) {
    front = 0;
    rear = 0;
  } else {
    rear ++;
  }
  int val;
  printf("Enter the value: ");
  scanf("%d", &val);
  queue[rear] = val;
}

void dequeue() {
  if (front == -1 || front == rear) {
    printf("Queue Underflow\n");
  } else {
    int val = queue[front];
    front++;
    printf("The element deleted: %d\n", val);
  }
}

void display() {
  if (front == -1 || front == rear) {
    printf("Queue Underflow\n");
  } else {
    for (int i = 0; i < max; i++) {
      if (i<front || i>rear) {
        printf("NaN ");
      } else {
        printf("%d ", queue[i]);
```

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

int main() {
  int choice;
  int con = 1;

  while (con == 1) {
    printf(" 1.Enqueue()\n 2.Dequeue()\n 3.Display()\n");
    printf("Choose a operation: ");
    scanf("%d", &choice);
    switch (choice) {
    case 1:
      enqueue();
      break;
    case 2:
      dequeue();
      break;
    case 3:
      display();
      break;
    default:
      printf("Invalid choice");
      break;
    }
    printf("Enter 1 to continue: ");
    scanf("%d", &con);
  }
}
```

Output:

```
 1.Enqueue()
 2.Dequeue()
 3.Display()
Choose a operation: 1
Enter the value: 2

Enter 1 to continue: 1

 1.Enqueue()
 2.Dequeue()
 3.Display()
Choose a operation: 1
Enter the value: 3

Enter 1 to continue: 1

 1.Enqueue()
 2.Dequeue()
 3.Display()
Choose a operation: 3
2 3 NaN NaN NaN

Enter 1 to continue: 1

 1.Enqueue()
 2.Dequeue()
 3.Display()
Choose a operation: 2
The element deleted: 2

Enter 1 to continue: 1

 1.Enqueue()
 2.Dequeue()
 3.Display()
Choose a operation: 3
NaN 3 NaN NaN NaN
```