11. Write a C program to implement Stack operations such as PUSH , POP , PEEK .

```
#include <stdio.h>
#define SIZE 100

int stack[SIZE];
int top = -1;

// Push operation
void push(int value) {
   if (top == SIZE - 1) {
      printf("Stack Overflow! Cannot push %d\n", value);
   } else {
      top++;
      stack[top] = value;
      printf("Pushed %d into stack.\n", value);
   }
}

// Pop operation
void pop() {
   if (top == -1) {
      printf("Stack Underflow! Nothing to pop.\n");
   } else {
      printf("Popped %d from stack.\n", stack[top]);
      top--;
   }
}
```

```
// Peek operation
void peek() {
   if (top == -1) {
      printf("Stack is empty.\n");
   } else {
      printf("Top element is: %d\n", stack[top]);
   }
}

// Display stack
void display() {
   if (top == -1) {
      printf("Stack is empty.\n");
      return;
   }
   printf("Stack elements: ");
   for (int i = top; i >= 0; i--) {
      printf("%d ", stack[i]);
   }
   printf("\n");
}

int main() {
   int choice, value;

   do {
      printf("\n--- Stack Operations ---\n");
      printf("1. PUSH\n");
```

```
printf("2. POP\n");
printf("3. PEEK\n");
printf("4. DISPLAY\n");
printf("5. EXIT\n");
printf("Enter your choice: ");
scanf ("%d", &choice);

switch(choice) {
   case 1:
      printf("Enter value to push: ");
      scanf("%d", &value);
      push(value);
      break;
   case 2:
      pop();
      break;
   case 3:
      peek();
      break;
   case 4:
      display();
      break;
   case 5:
      printf("Exiting program.\n");
      break;
   default:
      printf("Invalid choice! Try again.\n");
}
```

```
    } while(choice != 5);


    return 0;

}
```

OUTPUT