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

#define MAX_SIZE 10
int stack[MAX_SIZE];
int top = -1;

void push(int value);
void pop();
void peek();
void display();

int main()
{
   int choice;
   int value;

   while (1)
   {
      printf("\n***** STACK MENU ******");
      printf("\n1. PUSH ");
      printf("\n2. POP ");
      printf("\n3. PEEK");
      printf("\n4. DISPLAY ");
      printf("\n5. EXIT\n");
      printf("Choice: ");

      if (scanf("%d", &choice) != 1) {
         while (getchar() != '\n');
         printf("\nInvalid input. Please enter a number.\n");
         continue;
      }

      switch(choice)
      {
         case 1:
            printf("Enter value: ");
            scanf("%d", &value);
            push(value);
            break;
         case 2:
            pop();
            break;
         case 3:
            peek();
            break;
         case 4:
            display();
            break;
         case 5:
            printf("\nExiting program. Goodbye!\n");
```

```
            exit(0);
        default:
            printf("\nInvalid Choice! Select 1-5.\n");
      }
   }

   return 0;
}

void push(int value)
{
   if (top >= MAX_SIZE - 1)
   {
      printf("\nCannot push %d. Stack is full.\n", value);
   }
   else
   {
      top++;
      stack[top] = value;
      printf("\n%d Pushed.\n", value);
   }
}

void pop()
{
   if (top == -1)
   {
      printf("\nStack is empty.\n");
   }
   else
   {
      int popped_value = stack[top];
      top--;
      printf("\n%d Popped.\n", popped_value);
   }
}

void peek()
{
   if (top == -1)
   {
      printf("\nStack is empty. Nothing to peek.\n");
   }
   else
   {
      printf("\nTop element is %d.\n", stack[top]);
   }
}

void display()
{
```

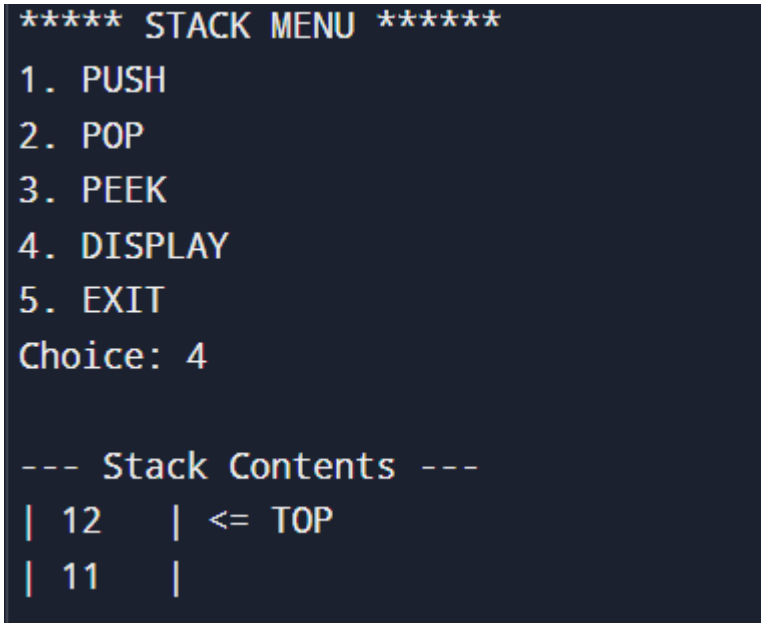
```
    if (top == -1)
    {
        printf("\nStack contents: [] (Empty)\n");
        return;
    }

    printf("\n--- Stack Contents ---\n");
    for (int i = top; i >= 0; i--)
    {
        printf("| %-4d |", stack[i]);
        if (i == top) {
            printf(" <= TOP");
        }
        printf("\n");
    }
}
```

```
***** STACK MENU ******
1. PUSH
2. POP
3. PEEK
4. DISPLAY
5. EXIT
Choice: 1
Enter value: 11

11 Pushed.

***** STACK MENU ******
1. PUSH
2. POP
3. PEEK
4. DISPLAY
5. EXIT
Choice: 1
Enter value: 12
```


***** STACK MENU ******
1. PUSH
2. POP
3. PEEK
4. DISPLAY
5. EXIT
Choice: 4
--- Stack Contents ---
| 12 | <= TOP
| 11 |