```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>     // For INT_MIN

#define SIZE 100

// Create a stack with capacity of 100 elements
int stack[SIZE];

// Initially stack is empty
int top = -1;
int i;


/* Function declaration to perform push and pop on stack */
void push(int element);
int  pop();
void display();


int main()
{
   int choice, data;

   while(1)
   {
      /* Menu */
      printf("------------------------------------\n");
      printf("1. Push\n");
```

```
printf("2. Pop\n");
printf("3. Size And Display\n");
printf("4. Exit\n");
printf("-----------------------------------\n");
printf("Enter your choice: ");

scanf("%d", &choice);

switch(choice)
{
   case 1:
      printf("Enter data to push into stack: ");
      scanf("%d", &data);

      // Push element to stack
      push(data);
      break;

   case 2:
      data = pop();

      // If stack is not empty
      if (data != INT_MIN)
         printf("Data => %d\n", data);
      break;

   case 3:
  display();
      printf("Stack size: %d\n", top + 1);
```

```
            break;

        case 4:
            printf("Exiting from app.\n");
            exit(0);
            break;

        default:
            printf("Invalid choice, please try again.\n");
      }

      printf("\n\n");
   }


   return 0;
}



/**
 * Functiont to push a new element in stack.
 */
void push(int element)
{
   // Check stack overflow
   if (top >= SIZE)
   {
      printf("Stack Overflow, can't add more element element to stack.\n");
```

```
        return;
    }

    // Increase element count in stack
    top++;

    // Push element in stack
    stack[top] = element;

    printf("Data pushed to stack.\n");
}



/**
 * Function to pop element from top of stack.
 */
int pop()
{
    // Check stack underflow
    if (top < 0)
    {
        printf("Stack is empty.\n");
        return INT_MIN;
    }



    // Return stack top and decrease element count in stack
    return stack[top--];
}
```

```
/**
 * Function to dispaly element from the stack.
 */
void display()
{
 if(top >= 0)
 {
   //  Print the stack
    printf("\nELEMENTS IN THE STACK\n\n");
    for(i=top;i>=0;i--)
         printf("%d\t",stack[i]);
 }
 else
 {
      printf("\nEMPTY STACK\n");
 }
 }
```

```
-------------------------------------
1. Push
2. Pop
3. Size And Display
4. Exit
-------------------------------------
Enter your choice: 2
Data => 6


-------------------------------------
1. Push
2. Pop
3. Size And Display
4. Exit
-------------------------------------
Enter your choice: 3

ELEMENTS IN THE STACK

8       5       Stack size: 2


-------------------------------------
1. Push
2. Pop
3. Size And Display
4. Exit
-------------------------------------
Enter your choice: 4
Exiting from app.


-----------------
(program exited with code: 0)

Press any key to continue . . .
```