Program:

```
#include <stdio.h>
#define MAXSIZE 10

struct stack
{
   int Stack_array[MAXSIZE];
   int top;
};

typedef struct stack STACK;
STACK s;

void push(void);
void pop(void);
void display(void);
void peek(void);
void size(void);
void isEmpty(void);
void isFull(void);

int main()
{
   s.top = -1;
   int choice;
   int option;

   printf("Implementing Stack ADT using Arrays \n");
   printf("What operation do you want to perform \n");
   printf(" For PUSH enter 1 \n");
   printf(" For POP enter 2  \n");
```

```
printf(" For DISPLAY  enter 3 \n");
printf(" To see TOPMOST element  enter 4 \n");
printf(" To find the stack Size enter 5 \n");
printf(" To check if Stack is EMPTY enter 6 \n");
printf(" To check if Stack is FULL enter 7 \n");
do
{

   printf("Enter your choice\n");
   scanf("%d", &choice);
   switch (choice)
   {
   case 1:
      push();
      break;
   case 2:
      pop();
      break;
   case 3:
      display();
      break;
   case 4:
      peek();
      break;
   case 5:
      size();
      break;
   case 6:
      isEmpty();
      break;
   case 7:
```

```
            isFull();
            break;
        }

        printf(" \n Do you want to continue(Type 0 for no and 1 for yes)?\n");
        scanf("%d", &option);

    } while (option == 1);
    return 0;
}

void push()
{
    int num;
    if (s.top == (MAXSIZE - 1))
    {
        printf("Stack Overflow\n");
    }
    else
    {
        printf("Enter the element to be pushed\n");
        scanf("%d", &num);
        s.top = s.top + 1;
        s.Stack_array[s.top] = num;
    }
}

void pop()
{
    int num;
    if (s.top == -1)
```

```
    {
        printf("Stack Underflow\n");
    }
    else
    {
        num = s.Stack_array[s.top];
        printf("The popped element is = %d ", s.Stack_array[s.top]);
        s.top = s.top - 1;
    }
}

void display()
{
    int i;
    if (s.top == -1)
    {
        printf("Stack is empty\n");
    }
    else
    {
        printf("\n The elements of the stack are \n");
        for (i = s.top; i >= 0; i--)
        {
            printf("%d\n", s.Stack_array[i]);
        }
    }
    printf("\n");
}

void peek()
{
```

```
    if (s.top == -1)
    {
        printf("Stack is empty\n");
    }
    else
    {
        printf("Topmost element is = %d", s.Stack_array[s.top]);
    }
}

void isEmpty()
{
    if (s.top == -1)
        printf("Stack is Empty\n");
    else
        printf("Stack is Not Empty\n");
}

void isFull()
{
    if (s.top == MAXSIZE - 1)
        printf("Stack is Full\n");
    else
        printf("Stack is Not Full\n");
}

void size()
{
    printf("The Stack size is %d \n", s.top + 1);
}
```

Output:

Implementing Stack ADT using Arrays

What operation do you want to perform

For PUSH enter 1

For POP enter 2

For DISPLAY enter 3

To see TOPMOST element enter 4

To find the stack Size enter 5

To check if Stack is EMPTY enter 6

To check if Stack is FULL enter 7

Enter your choice

1

Enter the element to be pushed

5

Do you want to continue(Type 0 for no and 1 for yes)?

1

Enter your choice

1

Enter the element to be pushed

4

Do you want to continue(Type 0 for no and 1 for yes)?

1

Enter your choice

3

The status of the stack is

4

5

Do you want to continue(Type 0 for no and 1 for yes)?

1

Enter your choice

5

The Stack size is 2

Do you want to continue(Type 0 for no and 1 for yes)?

1

Enter your choice

3

The status of the stack is

4

5

Do you want to continue(Type 0 for no and 1 for yes)?

1

Enter your choice

4

Topmost element is = 4

Do you want to continue(Type 0 for no and 1 for yes)?

0