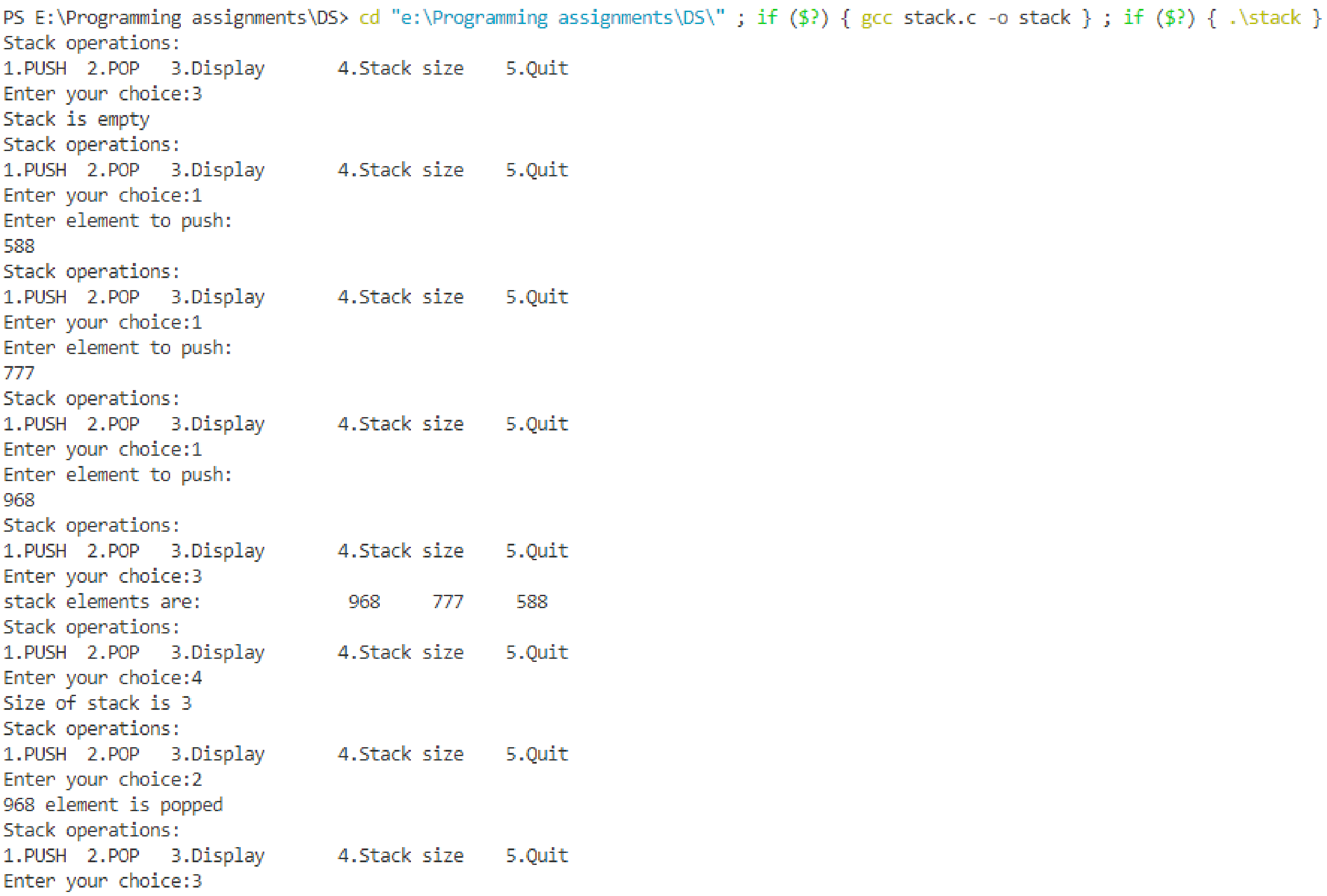
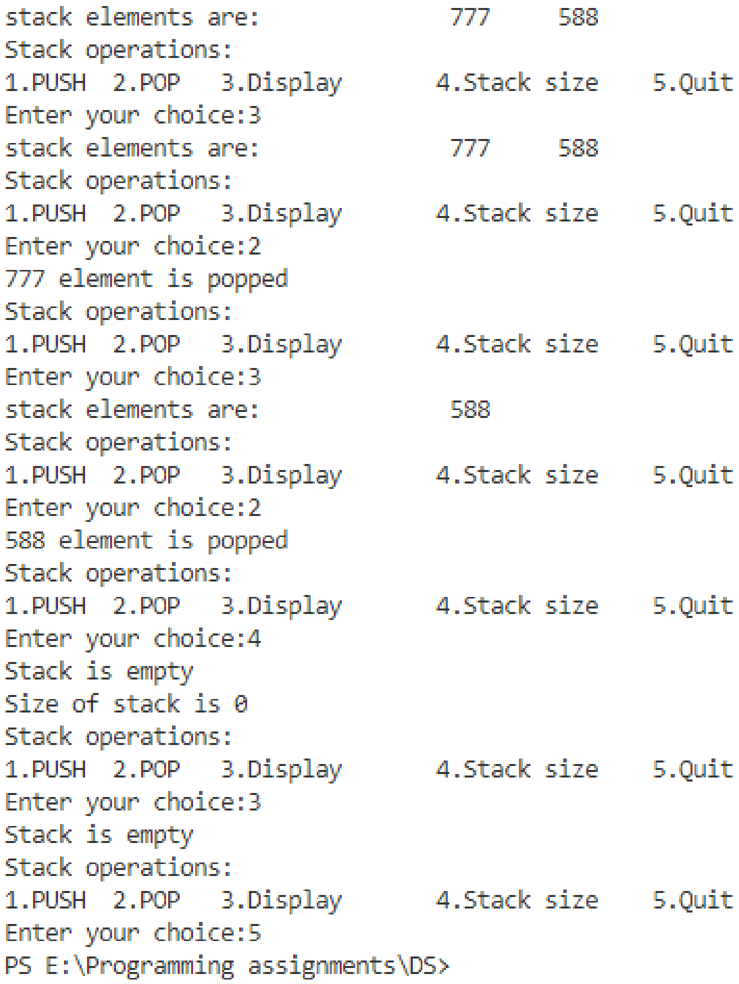


```
PS E:\Programming assignments\DS> cd "e:\Programming assignments\DS\" ; if ($?) { gcc stack.c -o stack } ; if ($?) { .\stack }
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:3
Stack is empty
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:1
Enter element to push:
588
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:1
Enter element to push:
777
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:1
Enter element to push:
968
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:3
stack elements are:              968     777     588
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:4
Size of stack is 3
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:2
968 element is popped
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:3
```



```
stack elements are:              777     588
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:3
stack elements are:              777     588
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:2
777 element is popped
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:3
stack elements are:              588
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:2
588 element is popped
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:4
Stack is empty
Size of stack is 0
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:3
Stack is empty
Stack operations:
1.PUSH  2.POP   3.Display       4.Stack size    5.Quit
Enter your choice:5
PS E:\Programming assignments\DS>
```

```
int length(){
    int i, count = 0;
    if (isempty())
        printf("Stack is empty\n");
    else
    {
        for (i = 0; i <= top; i++)
        count++;
    }
    return count;
}
int main(){
    int ch, ele, len = 0;
    do{
        printf("Stack operations:");
        printf("\n1.PUSH");
        printf("\t2.POP");
        printf("\t3.Display");
        printf("\t4.Stack size");
        printf("\t5.Quit");
        printf("\nEnter your choice:");
        scanf("%d", &ch);
        switch (ch){
            case 1:
                printf("Enter element to push: \n");
                scanf("%d", &ele);
                push(ele);
                break;
            case 2:
                ele = pop();
                printf("%d element is popped\n", ele);
                break;
            case 3:
                display();
                break;
            case 4:
                len = length();
                printf("Size of stack is %d\n", len);
                break;
            case 5:
                return 1;
            default:
                printf("Invalid choice\n");
            }
    }while (ch != 5);
    return 0;
}
```

Code :

```
#include <stdio.h>
#include <stdlib.h>
#define size 5
int stack[size];
int top = -1;
int isfull(){
    if (top == size - 1)
        return 1;
    else
        return 0;
}
int isempty(){
    if (top == -1)
        return 1;
    else
        return 0;
}
void push(int data){
    if (isfull())
        printf("Stack is full\n");
    else{
        top++;
        stack[top] = data;
    }
}
int pop(){
    int e;
    if (isempty())
        printf("Stack is empty\n");
    else{
        e = stack[top];
        top--;
    }
    return e;
}
void display(){
    int i;
    if (isempty())
        printf("Stack is empty\n");
    else{
        printf("stack elements are:\t");
        for (i = top; i >= 0; i--)
            printf("\t %d", stack[i]);

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