the following elements 50, 25, 37, 45, 55 into the

Exp (11): Write a C program to implement stack operations such as PUSH, POP and PEEK.

Aim: To write a C program to implement stack operations such as PUSH, POP and PEEK.

Algorithm:

*. Start.
*. Initialize stack and top.
*. Implement PUSH, POP, and PEEK operations.
*. Use menu to perform operations.
*. Stop.

program:

```
#include <stdio.h>
# define SIZE 5
  int stack[SIZE];
  int top = -1;
   void push() {
        int x;
        if (top == SIZE -1)
          printf("stack overflow\n");
        else {
           printf("Enter value to push: ");
            scanf("%d", &x);
            stack[++top] = x;
            printf("%d pushed\n", x);
          }
      }
     void pop(){
         if (top == -1)
            printf("stack underflow\n");
         else
           printf("popped: %d\n", stack[top--]);
       }
       void peek() {
           if (top == -1)
             printf("stack is Empty\n");
           else
             printf("TOP element : %d\n", stack[top]);
       }
```

```
void display () {
    if (top == -1)
        printf("stack is empty\n");
    else {
        printf("stack: ");
        for (int i = top; i >= 0; i--)
            printf("%d ", stack[i]);
        printf("\n");
    }
}
int main (){
    int choice;
    while(1){
    printf("\n1. push\n2. pop\n3. peek\n4. display
        \n5. exist\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);
    switch (choice) {
        case 1 : push(); break;
        case 2 : pop(); break;
        case 3 : peek(); break;
        case 4 : display(); break;
        case 5 : return 0;
        default : printf("Invalid choice\n");
    }
}
```

Output:

```
1. push, 2. pop, 3. peek, 4. display, 5. exit
Enter your choice: 1
Enter value to push: 34
34 pushed.
```

Result: Thus, the program executed successfully