Program

```c
#include<stdio.h>
#include<stdlib.h>
#define size 5
int stack[size],top=-1;
void push()
{
    int item;
    if(top==size-1)
      printf("\nstack overflow");
    else
      printf("\nenter element to be added:");
      scanf("%d",&item);
      top+=1;
      stack[top]=item;
}

void pop()
{
    if(top==-1)
      printf("\nstack underflow");
    else
      printf("popped element is %d",stack[top]);
      top-=1;
}

void peek()
{
    if(top==-1)
      printf("\nstack underflow\n");
    else
    printf("\nValue at the top of stack is: %d", stack[top]);
}

void display()
{
    int i;
    if(top==-1)
      printf("\nstack underflow\n");
    else
      printf("\n\n<<<<<<<<<<<<ELEMENTS IN STACK>>>>>>>>>>>>>\n");
      for(i=top;i>=0;i--)
        printf("%d\n",stack[i]);
}

int main()
{
    int item,ch;
    while (1)
    {
        printf("\n\n...........stack operations...........\n");
        printf("1.Push\n2.Pop\n3.Peek\n4.Display\n5.Exit");
        printf("\nEnter the choice: ");
        scanf("%d", &ch);

        switch(ch)
        {
          case 1:
            push();
            break;
          case 2:
            pop();
```

Experiment No: 5
Date:

STACK IMPLEMENTATION USING ARRAY

Aim:

Write a C program to implement stack using array with functions push, pop, peek and display

Algorithm:

1. Start
2. Define size 5
3. Create stack[size] and initialize top=-1
4. Create function push(item)

```
push(item)
    if (top == size-1)
        display ("overflow")
    else
        print (‘‘Enter the element’’)
        read(item)
        top = top+1
        Stack [top] = item
        print (‘‘element added’’)
    end if
end function
```

.

5. Create function pop()

```
pop()
    if (top == -1)
        display ("Underflow")
    else
        item=stack[top]
        top-top-1
    end if
end function
```

6. Create funtion peek()

```
            break;
          case 3:
            peek();
            break;
          case 4:
            display();
            break;
          case 5:
            exit(0);
        }
    }
    return 0;
}
```

Output

```
.............STACK OPERATIONS.............
1. Push
2. Pop
3. Peek
4. Display
5. Exit

enter your choice: 1
enter element to be added: 1
element inserted.

.............STACK OPERATIONS.............
1. Push
2. Pop
3. Peek
4. Display
5. Exit

enter your choice: 1
enter element to be added: 2
element inserted.

.............STACK OPERATIONS.............
1. Push
2. Pop
3. Peek
4. Display
5. Exit

enter your choice: 1
enter element to be added: 3
element inserted.

.............STACK OPERATIONS.............
1. Push
2. Pop
3. Peek
4. Display
5. Exit

enter your choice: 4

<<<<<<<STACK ELEMENTS>>>>>>>>>
3
```

```
peek()
   if (top ==-1)
      display (‘‘Underflow’’)
   else
      print(stack [top])
   end if
end function.
```

7. Create function display()

```
display()
    if (top==-1)
       display (‘‘underflow’’)
    else
       print (‘‘Elements in Stack’’)
       for (i=top to i>=0)
           print (stack [i])
           i - -
    end if
end function
```

8. Create main function

```
main()
    while true
        display the operations "1.push 2.pop 3.display 4.peek 5.exit"
        read the choice from the user
        create a switch case for the choice
            if case=1
                call push fuction
                break
            if case=2
                call pop function
                break
            if case=3
                call display function
                break
            if case=4
                call peek function
                break
            if c=5
                exit(0)
end function
```

9. Stop

```
2
1
.............STACK OPERATIONS.............
1. Push
2. Pop
3. Peek
4. Display
5. Exit

enter your choice: 2
popped element is 3

.............STACK OPERATIONS.............
1. Push
2. Pop
3. Peek
4. Display
5. Exit

enter your choice: 4

<<<<<<<STACK ELEMENTS>>>>>>>>>
2
1
.............STACK OPERATIONS.............
1. Push
2. Pop
3. Peek
4. Display
5. Exit

enter your choice: 3
Value at top of the stack is 2
.............STACK OPERATIONS.............
1. Push
2. Pop
3. Peek
4. Display
5. Exit

enter your choice: 5
```

Result:

Program has been executed successfully and obtained the output