URK2AI1092
KENNETH ANDERSON

EX NO: 1	**STACK IMPLEMENTATION USING STACK**
DATE: 09/01/2023	

AIM:

TO IMPLEMENT STACK USING ARRAY

ALGORITHM:

Push :

Step 1 - Check whether stack is FULL. (top SIZE-1)

Step 2 - If it is FULL, then display "Stack is FULL!!! Insertion is not possible!!!" and terminate the function.

Step 3 - If it is NOT FULL, then increment top value by one (top++) and store data

into stack Pop:

Step 1 - Check whether stack is EMPTY. (top -1)

Step 2 - If it is EMPTY, then display "Stack is EMPTY!!! Deletion is not possible!!!" and terminate the function. else blete empty top ===)

Step 3 - If it is NOT EMPTY, then delete top of the stack and decrement top value by one (top--).

Display:

Step 1 - Check whether stack is EMPTY. (top == -1)

Step 2 - If it is EMPTY, then display "Stack is EMPTY!!!" and terminate the

function. Step 3 - If it is NOT EMPTY, then print the stack.

Peek :

Step 1 - Check whether stack is EMPTY. (top == -1)

Step 2 - If it is EMPTY, then display "Stack is EMPTY!!! and terminate the

function. Step 3 - If it is NOT EMPTY, then return the top of the stack.

DESCRIPTION:

Push :

Step 1 - Check whether stack is FULL. (top SIZE-1)

Step 2 - If it is FULL, then display "Stack is FULL!!! Insertion is not possible!!!" and terminate the function.

Step 3 - If it is NOT FULL, then increment top value by one (top++) and store data into

stack Pop:

Step 1 - Check whether stack is EMPTY. (top -1)

Step 2 - If it is EMPTY, then display "Stack is EMPTY!!! Deletion is not possible!!!" and terminate the function. else blete empty top ===)

Step 3 - If it is NOT EMPTY, then delete top of the stack and decrement top value by one (top--).

Display:

Step 1 - Check whether stack is EMPTY. (top == -1)

Step 2 - If it is EMPTY, then display "Stack is EMPTY!!!" and terminate the

function. Step 3 - If it is NOT EMPTY, then print the stack.

Peek :

Step 1 - Check whether stack is EMPTY. (top == -1)

Step 2 - If it is EMPTY, then display "Stack is EMPTY!!! and terminate the

function. Step 3 - If it is NOT EMPTY, then return the top of the stack.

SOURCE CODE:

```
class cse :
 def __init__(self):
   self.stack =list()
   self.maxsize = 5
   self.top = -1

 def push(self,data): if self.top
  == self.maxsize -1:
  print("stack is full")
  else:
      self.top = self.top+1
      self.stack.append(data)
 def pop(self):
   if self.top == -1: print ("stack is empty , we
     cant perform pop
   operation") else:
     item = self.stack.pop()
     self.top = self.top-1
     print("deleted item:",item)
 def display (self):
   print(self.stack)

Icse = cse()
while(1):
 print("1.push") print("2.pop")
 print("3.display") print("4.exit")
 choice =int(input("enter your choice"))
 if choice==1: value=int(input("enter
 the data"))

   Icse.push(value)
 elif choice==2:
   Icse.pop()
 elif choice==3:
   Icse.display()
 else: break
```

OUTPUT:

```
1.push
2.pop
3.display
4.exit
enter your choice1
enter the data2
1.push
2.pop
3.display
4.exit
```

```
enter your choice1
enter the data7
1.push
2.pop
3.display
4.exit
enter your choice2
deleted item: 7
1.push
2.pop
3.display
4.exit
enter your choice3
[2]
1.push
2.pop
3.display 4.exit
enter your choice4
```

RESULT:
The program has been executed sucessfully