02/06/22

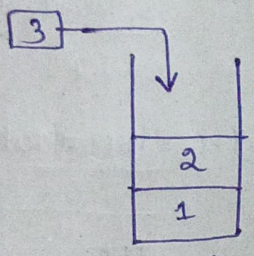
Module-2

STACK

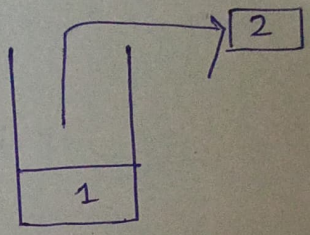
T.C = O(1)

- It is defined as a linear data structure that stores similar type of data items just like an array in a Last in First out (LIFO) way.
- In STACK insertion operation is technically called as 'PUSH' and deletion operation is technically called as 'POP'.
- In stack insertion and deletion of elements are performed at a single position which is known as 'top', that means a new element added top the stack and an element removed from the top of the stack.

ex :-



push



POP

Operations on stack :-

1. push() - To insert an element on top of the stack. If the stack is full then overflow cond'n is occurs.
2. pop() - To remove an element on top of the stack. If the stack is empty then it is called as underflow state.
3. isempty() - check whether the stack is empty or not.
4. isFull() - check whether the stack is full or not.
5. peek() - To print top of the element.
6. display() - To print all the element present in the stack.
7. count() - It counts total no. of elements present in the stack.

Push operation :-

The process of inserting a new element/data onto the stack is called push operation.

Step 1

Check whether the stack is full

Step-2 If the stack is full, then we can't insert the element, stack overflow condition is occurs. So produces an error and exit.

if (top == SIZE-1)

Step-3

If the stack is not full then increase the size of top. (top++).

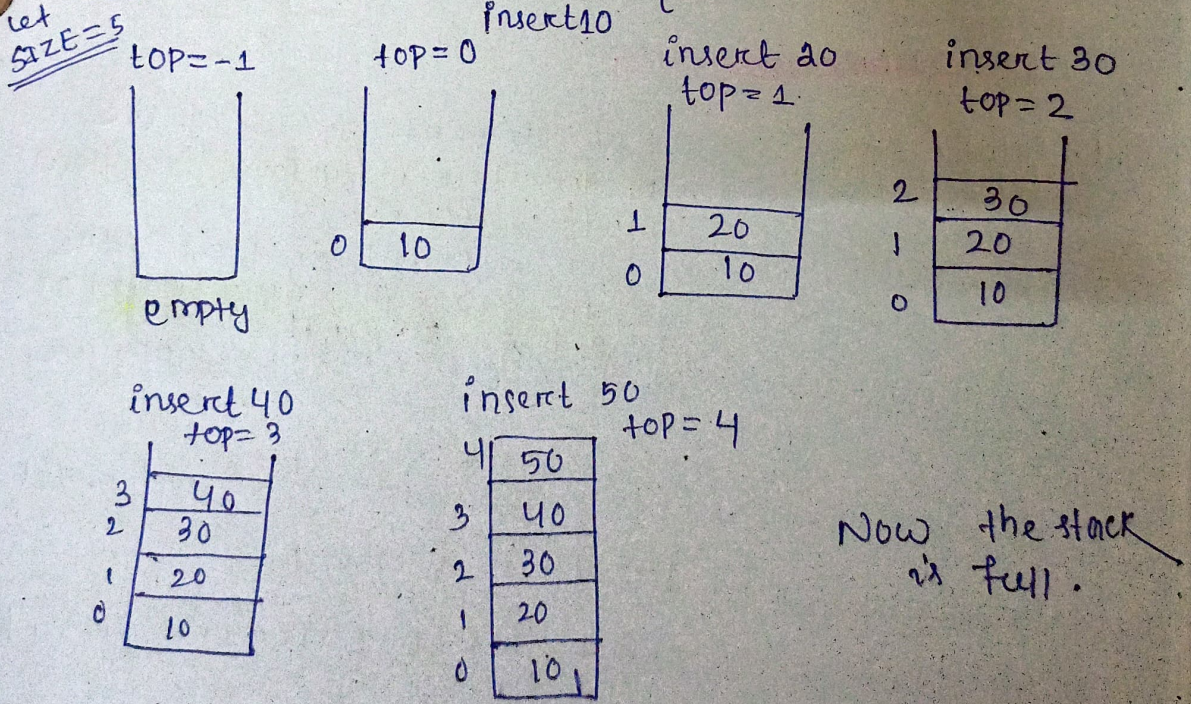
Step-4

Now the data element is placed, where top is pointing. stack[top] = val;

Step-5

The element will be inserted until the stack is full.

end:- print the is inserted successfully.



Now the stack is full.

POP operation :-

Deleting an element from the stack is called as POP operation.

Step-1

check whether the stack is empty. (top==-1)

Step-2

If the stack is empty then underflow condn is occurs. So it produces an error and exit.

Step-3

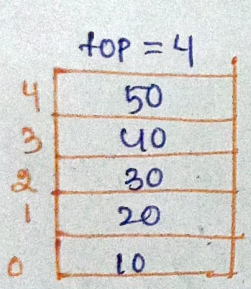
If the stack is not empty then print the element, which is present top of the stack.

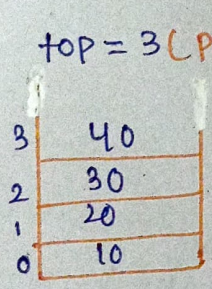
Step-4

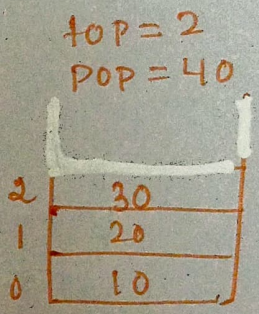
Now decrease the top size. (top--).

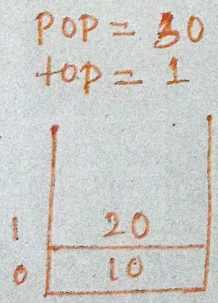
Step-5

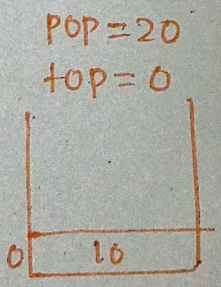
Return success

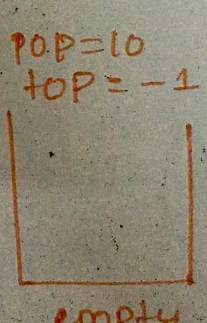












```
#include<stdio.h>
#include<stdlib.h>
 #define 6
int size[stack], top=-1;
//prototype
void push(int);
void pop();
void peek();
void display();
//push operation:-
void push(int val)
{
if (top==size-1)
  {
  printf("stack is full insertion not
                possible\n");
  }
else
  {
  top++;
  printf("The %d value inserted successfully
        into the stack \n", val);
  }
}
//pop operation :-
void pop()
  {
if(top==-1)
  {
  printf("stack is empty deletion not
                possible\n");
  }
else {
  printf("The %d value deleted successfully
                into the stack \n", stack[top]);
  top--;
} }
```

// peek operation :-

```
void peek()
{
if (top==-1)
{
printf("There is nothing in the top \n");
}
else{
printf("The top value of the stack is %d\n",
                                stack[top]);
}
}
```

// display Operation,

```
void display()
{ int p;
if( top== -1)
{
printf("stack is empty \n");
}
else{
for( p= top; p>= 0; p--)
{
printf("%d\n", stack[p]);
}
}
```

// main fun)

```
int main()
{
int ele,choice;
printf("select any one\n");
while(1)
{
```

```
printf("stack operation\n");
printf("1.push\n 2.pop\n 3.peek()\n
        4. display\n 5. exit(0));
  scanf("%d", &choice);
switch(choice){
case 1:
      printf("Enter the element to insert
              the stack\n");
      scanf("%d", &ele);
      push(ele);
       break;

    case 2:
       pop();
       break;

      case 3:
          peek();
          break();

          case 4:
           display();
           break();

              case 5::
                exit(0);

          }
    }
  return 0;

}
```