Stack: It is defined as a list of elemente in which we can insert or delete The element only at the top of the stack.

eg. in our daily life, we use a stack of CD's, stack of books.

and we can take CD or book only from the top.

similarly we can keep one more CD or book only at the top.

There are two operations possible on The stack.

O Push, when we add the item in stack.

@ Pop, when we delete The 'etern from stack.

The elements which are added (pushed) at the last in stack can be deleted (popped) from The stack, so its behaviour in like last in first out. So, stack is also called LIFO data structure

We can implement stack in two ways -

(1) Array implementation of stack.
(11) winhed hist implementation of stack.

Array implementation of stack."

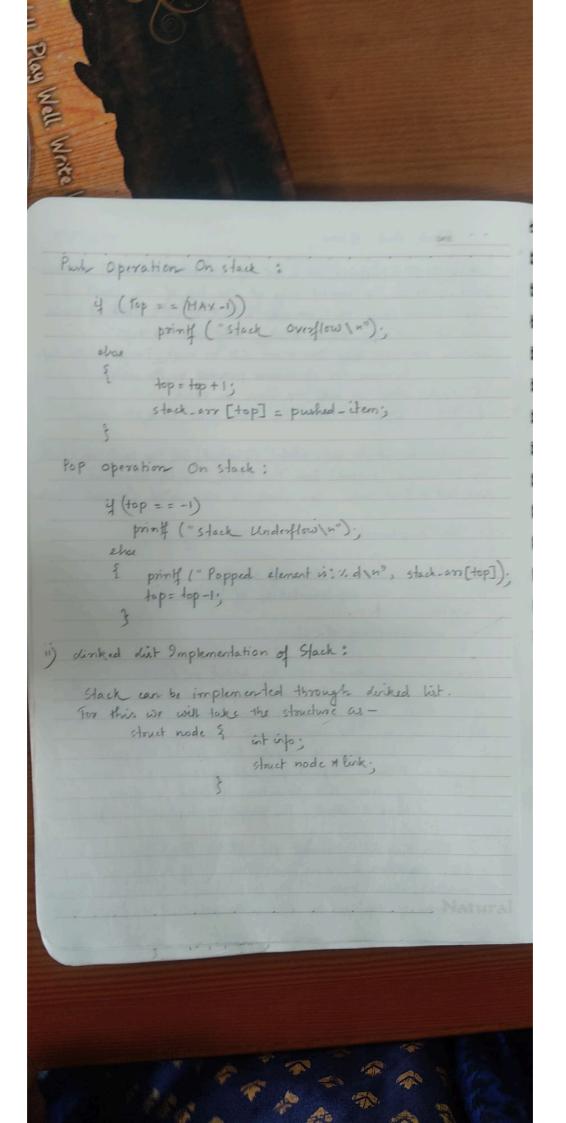
In array any element can be added or deleted at any place and we want to push or pop the element from top of the stack only. So we take a variable top, which keeps the position of the top element in array.

In Stack, we have to check two possibility cases.

1) when there is no place for adding (push) the element in the array; then the's condition is called overflow.

So, we have to check the value of top with size of an

When there is no element for deleting (pop) from stack, then this condition is called underflow. If there is no element in stack then value of top will be -1. So we have to choose the value of top before deleting the element of stack



Push Operation on stack: For pushing the element on stack we follow the insertion operation of the linked list, means we add the element at the start of the list. temp -> into = purhed - item; temp -> link = top; top = temp; Here top always points to the first mode of linked list. After statements, last pushed item will become first node of linked list. Since stack implementation is through linked list so we don't check overflow condition. Pop operation on Stack: For pop operation on stack we delete the first element of linked list because stack is a last in first out structure. of (Top == NULL) print (" stack is empty ("). else demp = top; printf (" Popped idem is 1/d \n", temp-rinfo) top = top -> link; free (temp). Here first we check for stack empty condition then we pop the first element of Stack."
Now top will point to second node of kinked hist which will become first node of Linked list. Natural

