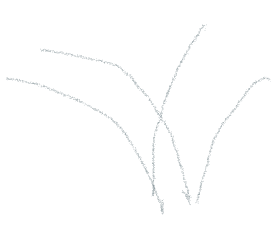
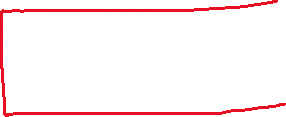
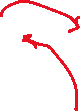
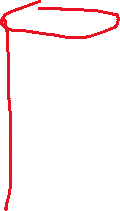
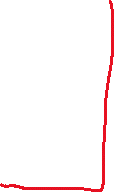
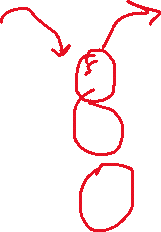
Stack:

1. Introduction
2. ADT stack

It works on last in first out.

It is collection of elements. In stack last element can be deleted first





Recursions:

Recursion are the functions which calls itself. It’s behaviiour is

Like a loop but they are internally used in stack.

Some recursion are directly convertible into iteration but some recursion are needed stack

And viseversa.

Note:

Every recursion can be converted into iteration. And every iteration can be conveted

Into recursion.

ADT stack:

It is the definition of stack in terms of data representation and operation.

Data:

1. Space for storing elements.
2. Top pointer.

Operations:

1. Push( ) : Inserting a value.
2. Pop( ) : Deleting a value
3. Peek(Index): looking at the value at the given position. It’s not deleting

.just knowing what is there.

1. StackTop( ): It’s not deleting just knowing what is at topmost value.
2. .isEmpty( ): Knowing wether stack is empty or stack is full.
3. .isFull( ):

In stack insertion and deletaion always takes place from the top

.inside the stack.

|  |
| --- |
|  |
| 15 |
| 20 |
| 16 |
| 10 |

4



3

2

1

0

We can store collection of elements –

1. Array
2. Linked-list.

Implementation of stack using array:

We need three things for implementation

* Fixed size array
* Size of the array
* Top pointer to point on the top pointing element, Data type should be integer.

Note:-

Inside the stack we insert and delete the element from same end.

Insertion is done from top and deletion is also done from top.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

0 1 2 3 4 5

|  |
| --- |
|  |
| 15 |
| 20 |
| 16 |
| 10 |

4

3

2

1

0

|  |
| --- |
|  |
|  |
|  |  |  |  |  |  |  |



struct stack

{

    int size;

    int Top;

    int \*s;

};

int main()

{

    struct stack st;

    cout<<"Enter the size of stack";

    cin>>st.size;

    st.s= new int[st.size];

    st.Top= -1;

}

Size = 5.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

4

3

2

1

0

-1 <---- Top

Condition for empty:-

. if( Top == -1)

Full:-

. if( Top == size -1)