Dynamic Stack – Till now we have implemented the stack using a static structure like array. However it can also be implemented using a dynamic structure like Linked List. Being dynamic, the stack can grow to any size (limited only by the available memory in the system). Hence there is no need to check the stack full condition.

Node structure to create a dynamic stack of int values.

struct node

{

int data;

struct node \*next;

};

Note that unlike in static stack where top is a position, here top is going to contain the address of topmost node. Hence declaration of top is not “int top”.

struct node \*top;

1. The push operation.
2. Push 10 in stack (1st element)

data next

Here next is NULL bcoz it is the 1st node in stack & has node

10 NULL

beneath it

newnode = 600

top = 600

1. Push 20

20 600

newnode->next = top;

10 NULL

newnode = 1024 top = newnode;

top = 1024

~~top~~ = 600

1. Push 30

30 1024

top = 854 You will observe that we are creating a LL only with

reverse links that is next node is connecting to

10 NULL

20 600

previous node & we are storing the address of

1024 last node (top most node)

600

1. Pop operation

30 1024

top = 854

10 NULL

20 600

int no;

no = top->data;

struct node \*temp = top;

top =1024 top = top->next; // top descends

free(temp);

return no;

600