**Aim:** Implement Stack ADT using an array.

#include <limits.h> #include <stdio.h> #include <stdlib.h> struct Stack {

int top;

unsigned capacity; int\* array;

};

struct Stack\* createStack(unsigned capacity)

{

struct Stack\* stack = (struct Stack\*)malloc(sizeof(struct Stack)); stack->capacity = capacity;

stack->top =-1;

stack->array = (int\*)malloc(stack->capacity \* sizeof(int)); return stack;

}

int isFull(struct Stack\* stack)

{

return stack->top == stack->capacity-1;

}

int isEmpty(struct Stack\* stack)

{

return stack->top ==-1;

}

void push(struct Stack\* stack, int item)

{

(isFull(stack)) return;

stack->array[++stack->top] = item; printf("%d pushed to stack\n", item);

}

int pop(struct Stack\* stack)

{

if (isEmpty(stack)) return INT\_MIN;

return stack->array[stack->top--];

}

int peek(struct Stack\* stack)

{

if (isEmpty(stack)) return INT\_MIN;

return stack->array[stack->top];

}

int main()

{

struct Stack\* stack = createStack(100); push(stack, 10);

push(stack, 20);

push(stack, 30);

printf("%d popped from stack\n", pop(stack)); return 0;

}