//CS311 Yoshii

//INSTRUCTION:

// Look for \*\* to complete the program (do control-S)

// Update the comments using the HOW TO COMMENT file,

// When done, compile stack.cpp to make sure there are no syntax errors.

// NOTE: functions are called instead of repeating

// the same lines of code.

//=========================================================

//HW#: HW1P1 stack

//Your name: Alexnader Sadeghipour

//Complier: g++

//File type: stack implementation file stack.cpp

//=========================================================

#include <iostream>

#include <stdlib.h>

#include <string>

#include "stack.h"

using namespace std;

stack::stack()

{

top = -1; // indicate an empty stack };

}

stack::~stack()

{

}

//PURPOSE: checks top and returns true if empty, false otherwise.

bool stack::isEmpty()

{ if (top == -1) return true; else return false;

}

//PURPOSE: checks top and returns true if full, false otherwise.

bool stack::isFull()

{ if (top == MAX-1 ) return true; else return false;

}

//PURPOSE: calls isFull and if true, throws an exception (Overflow)

// Otherwise, dds an element to el after incrementing top.

//PARAMETER: pass the element (elem) to be pushed

void stack::push(int x,}

{ if (isFull()) int x ;

else {top++; el[top] = elem;

}

//PURPOSE: calls isEmpty and if true, throws an exception (Underflow)

// Otherwise, removes an element from el and gives it back.

//PARAMETER: provide a holder (elem) for the element popped (pass by ref)

void stack::pop(el\_t& elem)

{ if (isEmpty()) ;

else {elem = el[top]; top--;}

}

// PURPOSE: calls isEmpty and if true, throws an exception (underflow)

// Otherwise, gives back the top element from el.

//PARAMETER: provide a holder (elem) for the element (pass by ref

void stack::push(int x}

{ if (isFull())int x;

cout<<"Stack overflow"

abort();

}

{

top++

arr[top] = x;

cout << x << "pushed to the top of the stack \n" ;

if (x<min)

min = x;

}

int main()