Exercise 14-1: The Median

#include <iostream>

#include <exception>

using namespace std;

void median(int numArray[], const int SIZE)

{

double result;

bool isSorted = true;

int j;

bool isZero = true;

for (int i = 0; i < SIZE; i++)

{

if (numArray[i] > 0)

{

isZero = false;

}

}

if (isZero || SIZE <= 0)

{

//throw numeric exception value of -1...

throw -1;

}

else

{

for (int i = 0; i < (SIZE - 1); i++)

{

if (numArray[i] > numArray[i + 1])

{

isSorted = false;

//throw numeric exception value of -2...

throw - 2;

}

}

if (isSorted)

{

//calculate median...

if (SIZE % 2 == 0)

{

//calculate even number median...

int a = numArray[SIZE / 2];

int b = numArray[(SIZE / 2) - 1];

result = (a + b) / (double)2;

}

else

{

//calculate odd number median...

result = numArray[SIZE / 2];

}

}

cout << endl;

}

cout << "The median is " << result << endl << endl;

}

void main() {

const int SIZE = 10; //adjustable...

int numArray[SIZE];

try

{

//POPULATE ARRAY...

cout << "Enter " << SIZE << " numbers from lowest to highest: " << endl;

for (int i = 0; i < SIZE; i++) {

cin >> numArray[i];

}

//find median for array...

median(numArray, SIZE);

}

catch (int a)

{

if (a == -1)

{

cout << "Error: No elements in array.\n";

}

else if (a == -2)

{

cout << "Error: Array not in sorted order.\n";

}

}

system("pause");

}

Exercise 14-2: Common User Error

#include <iostream>

#include <stdlib.h>

#include <windows.h>

#include <string>

using namespace std;

int getNum(char checker[], char x)

{

int flag = 0;

for (int i = 0; i < 10; i++)

{

if (x == checker[i]) //checks if input is digit...

{

return x;

flag = 1;

}

}

if (flag == 0)

{

throw - 1; //throw to catch in main...

}

}

void main() {

char x;

char checker[10] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' }; //digit checker...

int count = 10; //seconds of wait time when incorrect input is entered...

string quitEntry; //string ensures multiple characters can be

//entered without repeating the wait timer...

do

{

cout << "Enter a number: ";

try

{

cin >> x;

cin.clear();

cin.ignore();

x = getNum(checker, x);

}

catch (int) //I could have required user to input integer then end the program, but instead included a quit phrase and timer for every invalid entry...

{

system("cls");

while (count > 0)

{

cout << "Error: not a valid number. Please try again in " << count << " seconds..." << endl;

Sleep(1000); //wait time for invalid entry...

count--;

system("cls");

x = 0;

}

count = 10;

cin.clear();

cin.ignore();

}

system("cls"); //clear screen to start quit option...

cout << "Enter 'quit' to quit: ";

cin >> quitEntry;

system("cls");

} while (quitEntry != "quit");

system("pause");

}