/\*gavin skehan

21440824

30/11/21\*/

#include <stdio.h> // declare all the libraries

#include <math.h>

#include <stdlib.h>

#include <time.h>

int setPasscode(int N); // declare your functions

void randomPasscodeSearch(int passcode, int N);

void orderedPasscodeSearch(int passcode, int N);

//void finalPasscode(int passcode, int N);

int num;

int N;

int value;

int attempts = 0;

main()

{

int timer = time(NULL);

srand(timer); // randomizer for numbers

setPasscode(9); // this sets the max values for the passcode

randomPasscodeSearch(num,9);

orderedPasscodeSearch(num,9);

setPasscode(99);

randomPasscodeSearch(num,99);

orderedPasscodeSearch(num,99);

setPasscode(999);

randomPasscodeSearch(num,999);

orderedPasscodeSearch(num,999);

setPasscode(9999);

randomPasscodeSearch(num,9999);

orderedPasscodeSearch(num,9999);

/\*setPasscode(824);

randomPasscodeSearch(num, 824);

orderedPasscodeSearch(num, 824);

\*/

}

int setPasscode(int N)

{

num = rand() % N; // create random passcode

printf("Max passcode size = %d\n", N);

printf("%d\n", num);

return num;

}

void randomPasscodeSearch(int num, int N){ // this function outputs a random code to try match the original

printf("Random Search. Passcode = %d.", num); // attempts to match original code

do {

value = rand() % (N + 1);

if (value != num) {

attempts++; // for each attempt , attempts get plus 1

}

else {

printf(" ", attempts);

}

} while (value != num);

printf("Found after %d attempts.\n", attempts);

}

void orderedPasscodeSearch(int passcode, int N) // function searches for the code in order staring at 0

{

printf("ordered Search. Passcode = %d.", num);

do {

value = 0;

if (value != num) { // if the new code is not the origianl code

value++;

}

else {

printf(" ", value);

}

} while (value == num);

printf("Found after %d attempts.\n", passcode + 1); // the ordered search will be the passcode plus 1

/\* {

void finalPasscode(int passcode, int N); // part 5

printf("Final passcode is %d", finalPasscode);

scanf\_s("%d", finalPasscode);

}

\*/

Text

Description automatically generated