Ejercicio\_STOCK

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

Ejercicio el Stock Control.

Una cadena comercial adquirio un equipo tecnologico llamado "STOCK CONTROL" el cual almacena mercancia y controla su venta al publico de acuerdo a sus fechas de caducidad.

Para dicho control se aplica la teoria de PILAS (Ultimo en entrar, Primero en salir).

PROBLEMA.

Escribe un programa que muestre lista de "N" productos, almacenados en el STOCK; asi como la lista inversa de estos, para identificar el producto que debe ser comercializado inmediatamente.

DATOS DE ENTRADA.

Deberas ingresar un valor "N" que represente la cantidad de productos en el STOCK, asi como una lista "P" que representa los productos.

DATOS DE SALIDA.

Deberas mostrar la lista de productos, de acuerdo a la teoria de "PILAS", en donde el primer producto representa el que se debe comercializar inmediatamente.

EJEMPLO DATOS DE ENTRADA.

4

10

8

9

7

EJEMPLO DATOS DE SALIDA.

7

9

8

10

*Fuente: Problema Personalizado por Profr. Gener Ku Pech*

Problema subido por: [generk](https://omegaup.com/profile/generk/)

| **ID** | **Status** | **Porcentaje** | **Penalty** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/Ejercicio_STOCK#problems/new-run) | | | | | | | | |
| 0c130fc4 | Respuesta correcta | 100.00% | 0 | cpp | 2.93 | 0.00 |  |  |
| **Envíos** | | | | | | | |

<https://omegaup.com/arena/problem/Ejercicio_STOCK#problems>

// stack::push/pop

#include <iostream> // std::cout

#include <stack> // std::stack

#include <stdio.h>

using namespace std;

int main ()

{

std::stack<int> mystack;

int N;

scanf("%d", &N);

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

int e;

scanf("%d", &e);

mystack.push(e);

}

// std::cout << "Popping out elements...";

while (!mystack.empty())

{

std::cout << mystack.top() << endl;

mystack.pop();

}

std::cout << '\n';

// system("pause");

return 0;

}