Problema A

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

**Descripción**

Uber ha comenzado a aprender acerca de subconjuntos. Dada una lista de números, Uber debe seleccionar un subconjunto que maximice el producto. Sin embargo, el profesor quiere saber solamente el máximo producto obtenido de tomar exactamente dos números de la lista. Por favor ayude a Uber a encontrar la respuesta de esta tarea.

**Entrada**

La entrada inicia con una línea con un entero NN que representa la cantidad de números en la lista. La segunda línea contiene NN números PiPi para 0<0< ii ≤N≤N separados por espacios.

**Salida**

Se debe imprimir el número con el máximo producto posible.

**Ejemplo**

| **Entrada** | **Salida** |
| --- | --- |
| 7  1 2 3 8 4 1 10 | 80 |
| 4  -2 -3 2 1 | 6 |
| 3  -1 -2 3 | 2 |
| 3  1 ‐5 3 | 3 |

**Límites**

* 1<N<1000001<N<100000
* −1000000000≤Pi≤1000000000−1000000000≤Pi≤1000000000

*Fuente: Estructuras de Datos Universidad Nacional de Colombia*

Problema subido por: [Christian Zuluaga](https://omegaup.com/profile/cmzuluagam/)

| **Enviado** | **GUID** | **Status** | **Porcentaje** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/Problema-A#problems/new-run) | | | | | | | | |
| 2016-11-02 03:06:32 | 284de9a0 | Respuesta correcta | 100.00% | cpp | 3.05 MB | 0.01 s |  |  |
| 2016-11-02 03:01:53 | 884ee296 | Respuesta parcialmente correcta | 42.86% | cpp | 3.10 MB | 0.01 s |  |  |
| 2016-11-02 02:48:03 | 5ac4884b | Respuesta parcialmente correcta | 42.86% | cpp | 3.13 MB | 0.01 s |  |  |
| **Envíos** | | | | | | | |

<https://omegaup.com/arena/problem/Problema-A#problems>

#include <iostream>

#include <stdio.h>

#define ll long long int

using namespace std;

int main() {

int N;

scanf("%d", &N);

ll arr[N];

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

scanf("%lld", &arr[i]);

}

ll max\_prod = 0;

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

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

if(i!= j) {

max\_prod = std::max(max\_prod, arr[i] \* arr[j]);

}

}

}

printf("%lld\n", max\_prod);

system("pause");

return 0;

}