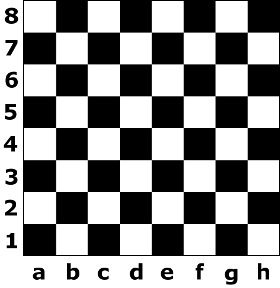
Tablero de Ajedrez

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

**Descripción**

Dadas las coordenadas de una casilla en un tablero de ajedrez de 8x8, determinar el color de dicha casilla.



**Entrada**

Una línea con la letra y dígito separados por espacios correspondientes a la coordenada de la casilla.

**Salida**

Imprime **BLANCO**, **NEGRO** según sea el caso.

**Ejemplos**

| **Entrada** | **Salida** | **Descripción** |
| --- | --- | --- |
| a 1 | Salida  NEGRO |  |

| **Entrada** | **Salida** | **Descripción** |
| --- | --- | --- |
| c 2 | Salida  BLANCO |  |

*Fuente: lacj20*

Problema subido por: [lacj20](https://omegaup.com/profile/lacj20/)

| **ID** | **Status** | **Porcentaje** | **Penalty** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/Tablero-de-Ajedrez#problems/new-run) | | | | | | | | |
| e322a53e | Respuesta correcta | 100.00% | 0 | cpp | 3.15 | 0.01 |  |  |
| **Envíos** | | | | | | | |

<https://omegaup.com/arena/problem/Tablero-de-Ajedrez#problems>

#include <iostream>

#include <stdio.h>

using namespace std;

int main() {

struct Helper{

char buffer[100];

std::string to\_string(int k){

sprintf(buffer, "%d", k);

return std::string(buffer);

}

};

char letra;

int dig;

scanf("%c %d", &letra, &dig);

Helper h;

string target = letra + h.to\_string(dig);

string tablero[8][8]; // = new string[8, 8];

string respuesta[8][8] =

{

{"BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO"},

{"NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO"},

{"BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO"},

{"NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO"},

{"BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO"},

{"NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO"},

{"BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO"},

{"NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO","NEGRO","BLANCO"},

};

int f = 0, c = 0;

for (int i = 8; i >= 1; i--)

{

for (int j = 'a'; j <= 'h'; j++)

{

tablero[f][ c] = ((char)j) + h.to\_string(i);// i.ToString();

c++;

}

f++;

c = 0;

}

int fila = 0, col = 0;

bool encontrado = false;

for (int i = 0; i < 8 && !encontrado; i++)

{

for (int j = 0; j < 8 && !encontrado; j++)

{

if (target == tablero[i][ j])

{

fila = i; col = j;

encontrado = true;

break;

}

}

}

cout << respuesta[fila] [col] << endl;

system("pause");

return 0;

}