

Añadir erase con iterador a unordered_map

Queremos añadir a la clase `unordered_map` un nuevo método

```
iterator erase(iterator const& it);
```

que dado un iterador apuntando a un par $\langle \text{clave}, \text{valor} \rangle$, elimine dicho par del diccionario y devuelva un iterador al siguiente par. Si el iterador recibido no apunta a ningún par, la operación produce un error. Para llevar a cabo esta operación no puede utilizarse la función `hash`.

Para probar esta nueva operación leeremos una serie de claves (números positivos) que se insertarán en un diccionario (asignándoles el valor 0, por ejemplo). Luego el diccionario se recorrerá con un iterador, borrando las claves que sean números pares. Después se recorrerá de nuevo el diccionario comprobando que todas las claves son números impares.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso ocupa dos líneas. La primera contiene el número $N > 0$ de claves que queremos insertar. En la segunda aparecen esas N claves, todas diferentes.

La entrada termina con un 0.

Salida

Si todo ha ido bien, se escribirá `TODO BIEN`. En otro caso, el programa abortará.

Entrada de ejemplo

```
5
1 2 3 4 5
5
1 11 111 1111 11111
5
2 22 222 2222 22222
6
1 3 5 2 4 6
0
```

Salida de ejemplo

```
TODO BIEN
TODO BIEN
TODO BIEN
TODO BIEN
```

Autor: Alberto Verdejo.