Añadir iteradores a `MapHash`

Los diccionarios no ordenados también pueden tener iteradores que permitan recorrer sus elementos. Naturalmente, no se puede esperar que los elementos sean recorridos en un orden concreto, ya que el diccionario no está ordenado, pero un iterador debe permitir recorrer todos los pares ⟨clave, valor⟩ pasando exactamente una vez por cada uno.

Añade a la clase `MapHash`, que implementa los diccionarios no ordenados mediante una *tabla hash abierta*, una clase interna `iterator` de iteradores y métodos `begin()` y `end()` para obtener iteradores que apunten al primer elemento en el recorrido y después del último.

Puedes sacar ideas de cómo están implementados los iteradores en la clase `ListLinkedDouble` de listas doblemente enlazadas o en la clase `MapTree` de diccionarios ordenados implementados mediante árboles binarios de búsqueda. Puedes simplificar y tener solamente iteradores no constantes, que permitan recorrer y modificar el diccionario. También, si lo necesitas, puedes simplificar la implementación de la clase `MapHash` para que no sea genérica, haciendo que tanto las claves como los valores sean enteros.

Para probar los iteradores vamos a construir un diccionario con claves y valores de tipo `int` y después vamos a recorrerlo calculando la suma de todos los valores asociados a claves impares. Este código se proporciona en la plantilla del problema.

Entrada

La entrada está formada por una serie de casos de prueba. Cada caso comienza por una línea con un número, la cantidad N de pares ⟨clave, valor⟩ que vamos a querer insertar en el diccionario ($1 \leq N \leq 100.000$). En las siguientes N líneas aparecen esos pares, primero la clave y luego el valor, separados por un espacio.

Salida

Para cada caso, se escribirá una línea con la suma de todos los valores asociados a claves impares, que habrá sido obtenida recorriendo el diccionario con un iterador.

Entrada de ejemplo

```
5
1 10
2 20
3 30
4 40
5 50
```

Salida de ejemplo

```
90
```