2) Otra función hash podría ser la siguiente:

Donde:

* m es la potencia de 2 que le sigue al tamaño de la tabla
* k es la clave a hashear
* h es el código hash
* A es una constante:

En este caso, nos queda de la siguiente manera:

Aplicando la fórmula a los valores dados, nos queda la siguiente tabla:

|  |  |
| --- | --- |
| **Valores** | **Hash** |
| 10168 | 43 |
| 21362 | 113 |
| 8515 | 143 |
| 7993 | 242 |
| 32434 | 80 |
| 29786 | 194 |
| 35175 | 88 |
| 50255 | 76 |
| 32305 | 150 |
| 24971 | 237 |
| 10177 | 187 |
| 53101 | 57 |
| 12901 | 65 |

Como se puede apreciar, quedaron 2 valores marcados en rojo porque exceden al tamaño de la tabla. Para ello tendríamos que dividirlos por 200 para obtener su resto. Por ende, la fórmula de hashing nos quedaría de la siguiente forma:

Nos quedaría la tabla de la siguiente forma:

|  |  |
| --- | --- |
| **Valores** | **Hash** |
| 10168 | 43 |
| 21362 | 113 |
| 8515 | 143 |
| 7993 | 42 |
| 32434 | 80 |
| 29786 | 194 |
| 35175 | 88 |
| 50255 | 76 |
| 32305 | 150 |
| 24971 | 37 |
| 10177 | 187 |
| 53101 | 57 |
| 12901 | 65 |

Ya no hay más colisiones.

Otra función hash que podríamos usar es el de plegado, el cual consiste en dividir por partes, sumar y aplicar el módulo, como dice la siguiente ecuación: