Report:

For m = 20000

	Hash1		Hash2	
	Number of collisions	**Average probes**	**Number of collisions**	**Average probes**
Chaining Method	2130	0.261	2144	0.26
Double Hashing	4157	0.376	3932	0.385
Custom Probing				

For m = 30000

	Hash1		Hash2	
	Number of collisions	**Average probes**	**Number of collisions**	**Average probes**
Chaining Method	1496	0.188	1474	0.158
Double Hashing	2249	0.224	2191	0.2
Custom Probing				

For m = 50000

	Hash1		Hash2	
	Number of collisions	**Average probes**	**Number of collisions**	**Average probes**
Chaining Method	916	0.106	966	0.099
Double Hashing	1135	0.106	1213	0.133
Custom Probing				

Hash function 1:

```
public int hashValue( String key, int tableSize ) {
    int hash = 23;
    char[] chars = key.toCharArray();

    for (char c : chars)
        hash = hash * 31 + c;

    hash = hash < 0 ? hash * -1 : hash;

    return hash % tableSize;
}
```

Hash function 2:

```java
public int hashValue( String key, int tableSize ) {

    int hash = 23;
    char[] chars = key.toCharArray();

    for (char c : chars)
        hash = (key.hashCode() + c) * 31 + 31;

    hash = hash < 0 ? hash * -1 : hash;

    return hash % tableSize;
}
```

Auxiliary hash function:

```java
public static int auxHashGenerator( String key, int
HASH_TABLE_SIZE ) {
    int hash = (key.hashCode() * 3) % HASH_TABLE_SIZE;
    hash = hash < 0 ? hash * -1 : hash;
    return hash;
}
```