Report on Offline: Hash Table

Student ID: 1805112

For Hash Table Size: 10009

	Hash 1		Hash 2	
	Number of collisions	**Average probes**	**Number of collisions**	**Average probes**
Chaining Method	3682	441/1000 = 0.441	3750	469/1000 = 0.469
Double Hashing	58260	5285/1000 = 5.285	59272	4514/1000 = 4.514
Custom Probing	60729	4717/1000 = 4.717	62547	7687/1000 = 7.687

For Hash Table Size: 20021

	Hash 1		Hash 2	
	Number of collisions	**Average probes**	**Number of collisions**	**Average probes**
Chaining Method	2147	245/1000 = 0.245	2092	243/1000 = 0.243
Double Hashing	3797	366/1000 = 0.366	3703	325/1000 = 0.325
Custom Probing	3889	355/1000 = 0.355	3727	313/1000 = 0.313

For Hash Table Size: 30013

	Hash 1		Hash 2	
	Number of collisions	**Average probes**	**Number of collisions**	**Average probes**
Chaining Method	1511	149/1000 = 0.149	1483	127/1000 = 0.127
Double Hashing	2207	198/1000 = 0.198	2180	172/1000 = 0.172
Custom Probing	2172	188/1000 = 0.188	2134	162/1000 = 0.162

Hash Function 1:

```
int Hash_Table::Hash_Function_1(string s)   // length 4 folding hash algorithm
{
    long long sum = 0, mul = 1;
    for (int i = 0; i < s.length(); i++)
    {
        mul = (i % 4 == 0) ? 1 : mul * 256;
        sum += s[i] * mul;
    }
    return (int)(abs(sum) % M);
}
```

Hash Function 2:

```
int Hash_Table::Hash_Function_2(string s)   // polynomial rolling hash algorithm
{
    const int p = 31;
    long long sum = 0;
    long long pow = 1;
    for (char c : s)
    {
        sum = (sum + (c - 'a' + 1) * pow) % M;
        pow = (pow * p) % M;
    }
    return (int)sum;
}
```

Auxiliary Hash Function:

```
int Hash_Table::aux_hash(string s)
{
    long long int sum = 0;
    for (char c : s)
    {
        sum = sum + ((c - 'a') * (c - 'a'));
    }
    return (int)(abs(sum) % M);
}
```