Name: Abu Nowshed Sakib

ID: 1705107

Dept: CSE, BUET

Course: CSE208

Topic: Hashing Offline

N = 10000

Collision Resolution Method	Hash1		Hash2	
	Number of Collisions	Average Probes	Number of Collisions	Average Probes
Chaining Method	3698	1.467	3689	1.466
Double Hashing	61337	6.82	54886	6.697
Custom Probing	57255	5.523	60438	6.405

N = 20000

Collision Resolution Method	Hash1		Hash2	
	Number of Collisions	Average Probes	Number of Collisions	Average Probes
Chaining Method	7325	1.398	7371	1.422
Double Hashing	124157	5.377	126815	3.98
Custom Probing	121985	5.034	122860	4.68

N = 50000

Collision Resolution Method	Hash1		Hash2	
	Number of Collisions	Average Probes	Number of Collisions	Average Probes
Chaining Method	18349	1.311	18369	1.292
Double Hashing	334900	1.628	340331	1.673
Custom Probing	334373	1.705	333293	1.665

N = 80000

Collision Resolution Method	Hash1		Hash2	
	Number of Collisions	Average Probes	Number of Collisions	Average Probes
Chaining Method	29614	1.194	29717	1.183
Double Hashing	763256	1.261	783496	1.271
Custom Probing	552713	1.285	600430	1.296

Hash Functions:

1st Hash function:

```
int hash1(string key)
{
    long long int hashIdx = 0;
    int max_prime = 1e9 + 9;
    for(int i=0; i<key.length(); i++)
    {
        hashIdx = ((hashIdx*37) + key[i])% max_prime;


    }
    return int(hashIdx%hashSize);
}
```

2nd Hash Function:

```
Int hash2(string key) ///polynomial rolling hash function
{
    long long int hashIdx = 0;


    int primeP = 31;
    long long int powerP = 1;
    int max_prime = 1e9 + 9;

    for(int i=0; i<key.length(); i++)
    {
        hashIdx = (hashIdx+ (key[i]-'a'+1)*powerP)%max_prime;
        powerP = (powerP*primeP)%max_prime;
    }
    return int(hashIdx%hashSize);
}
```

Aux Hash Function:

```
int auxHash(string key)
{
    long long int hashIdx = 0;
    for(int i=0; i<key.length(); i++)
    {
        hashIdx = hashIdx + key[i];
    }
    return int(hashIdx%hashSize);
}
```