README:

Splash Table is bucketized version of d-ary cuckoo hashing. This hash table mainly improves the probing over cuckoo hashing by removing control dependency from the probing routing and relying on data dependency thus reducing the number of cycles required to perform the probing of the hash-table. A splash table inserts an item using multiple hash functions. A key can be inserted into a bucket corresponding to any one of the functions. Probes need to consult all hash buckets for the key. The cool thing about splash tables is that if there is no room in any bucket for a new key, one can evict some other key to make room. This other key can be re-inserted elsewhere, depending on the values of its other hash functions.

**Part1: This part includes following files**

1. SplashTable.java: Main file containing the code for hash table.
2. KeyValue.java: Wrapper used for the key value pair.
3. Makefile: Make file to execute the program.
4. manifest: File used to create executable jar.
5. inputfile: Sample input file
6. dumpfile: Sample dump file
7. probefile: Sample probe file
8. resultfile: Sample result file

This can be run using a make file as follows:

All the files should be in the same directory including inputfile, dumpfile, probefile and resultfile. And java 1.7 and above should be installed in the system. To create the jar file java -cfm jar command has been used.

1. make : To build the classes and generate class files.
2. make jarfile: To create the executable jar file of the generated class files and name of the jar file will be "splash.jar".
3. make splash B=<value> R=<value> S=<value> h=<value> inputfile=<> dumpfile=<> probefile=<> resultfile=<> :

To run executable splash jar with various inputs. For this step inputfile, dumpfile, probefile and resultfile have been given defaults to be inputfile, dumpfile, probefile and resultfile respectively. And B, R, S, h have been given defaults to be 2, 1, 2 and 1 respectively.

1. make clean: To clean the class files.

This has been implemented in Java. Major components of this are following functions:

***1. build*** : This method performs insertions into the hashtable using the concept of reinsertion as stated above. In this we have used the first bucket with space concept to insert into a bucket and evict the item from one of the buckets if no space is available in any of the candidates obtained after hashing the key. Number of candidates depends upon the number of hash functions. After evict the first inserted element current element is inserted and then the process is repeated with the evicted element. For normal cases, insertion takes place in the corresponding bucket according to the hash function. If the number of reinsertions exceed the constant value specified by the user as an input it execution will stop and SplashTable will exit with necessary details in to the dumpfile.

***2. dump:*** This function dumps the current state of the hash table into a file whose name is entered by the user, else default name is dumpfile. If a bucket/ slot does not contain any entry, considering the fact as given in the project that 0 is not a valid key and value, corresponding entries are filled with zeroes. To indicate that they are empty. Else, it displays the values of B, S, h and N at the top in that order. Next it displays the odd multipliers used to create the hash functions. And then the entries of the buckets depending upon the size of the table.

***3. probe:*** This method is responsible for probing the hash table contents for a specific key from a given file and writes the results in to resultfile.