|  |  |
| --- | --- |
| **Data Structures & Algorithms**  Diploma in CST, IT  Year 2/3 (2020/21) Semester 4/6 | **Week 6** |
| **1-2 Hours** |
| **Tutorial 6 – Hash Tables** | |

1. Devise a *perfect* hash function for the following domains. What is the range of your function?
2. The set of integers in [-100, 100]

|  |
| --- |
|  |

1. The set of 3-letter English words

|  |
| --- |
|  |

1. The set of tic-tac-toe positions

|  |
| --- |
|  |

1. The specification of the Dictionary ADT implemented using a Hash Table with separate chaining is given below.

|  |
| --- |
| // Dictionary.h - - Specification of Dictionary ADT  #include<string>  #include<iostream>  using namespace std;  const int MAX\_SIZE = 100;  typedef string ItemType;  typedef int KeyType;  struct Node  {  KeyType key; // search key  ItemType item; // data item  Node \*next; // pointer pointing to next item  };  class Dictionary  {  private:  Node \*items[MAX\_SIZE];  int size; // number of items in the Dictionary  public:  // constructor  Dictionary();  // destructor  ~Dictionary();  int hash(KeyType key);  // add a new item with the specified key to the Dictionary  bool add(KeyType newKey, ItemType newItem);  // remove an item with the specified key in the Dictionary  void remove(KeyType key);  // get an item with the specified key in the Dictionary (retrieve)  ItemType get(KeyType key);  // check if a specified key is in the Dictionary  bool contains(KeyType key);  // check if the Dictionary is empty  bool isEmpty();  // check the size of the Dictionary  int getLength();    //------------------- Other useful functions -----------------  // display the items in the Dictionary  void print();  }; |

Implement the following operations of the Dictionary ADT

1. **int hash(KeyType key);**

|  |
| --- |
|  |

1. **bool add(KeyType key, ItemType item);**

|  |
| --- |
|  |

1. **bool remove(KeyType key);**

|  |
| --- |
|  |