Page No.
Date 25 11 20

MD YASEEN AHMED
1BM19CS404.

-> Program to implement functions of Dictionary using Hashing.

```
Class HashNode
{
  public:
    int Key;
    int value;
    HashNode* Next;

     HashNode(int Key, int value)
     {
       this -> key = Key;
       this -> value = value;
       this -> next = NULL;
     }
 };
```

```
class HashMap
{
  private:
     HashNode** htable;

  public:
     HashMap()
     {
        htable = new HashNode*[SIZE];

    for(int i=0; i<SIZE; i++)
           htable[i] = NULL;
     }

  ~HashMap()
  {
     for(int i=0; i<SIZE; ++i)
      {
         HashNode* entry = htable[i];
       while (entry != NULL)
         {
  HashNode* prev = entry;
       entry = entry->next;
```

```
            delete prev;
        }
    }
    delete[] htable;
}

int HashFunction(int Key)
{
    return Key % SIZE;
}

void insert(int Key, int value)
{
    int hash_val = HashFunction(Key);
    HashNode* prev = NULL;
    HashNode* entry = htable[hash_val];
    while(entry != NULL)
    {
        prev = entry;
        entry = entry->next;
    }
    if (entry == NULL)
    {
```

```
    entry = new HashNode(Key, value);
     if (prev == NULL)
     {
        htable[hash_val] = entry;
     }
    else
     {
        prev->next = entry;
      }
    }
    else
    {
       entry->value = value;
     }
  }

void Remove (int Key)
{
    int hash_val = Hashfunction (key);
    HashNode* entry = htable[hash_val];
    HashNode* prev = NULL;
```

```
if (entry == NULL || entry ->key != key)
{
   cout << "No Element Found at key";
   return;
}
 while (entry -> next != NULL)
  {
     prev = entry;
     entry = entry ->next;
  }
   if (prev != NULL)
      prev -> next = entry ->next;

   delete entry;
   cout << "Element Deleted ";
}

int search (int key)
{
   bool flag = false;
   int hash_val = HashFunc(key);
   HashNode* entry = htable[hash_val];
```

```
while(entry != NULL)
{
    if (entry -> key == key)
    {
        cout << entry -> value << "  ";
         flag = true;
     }
     entry = entry -> next;
  }
  if (!flag)
        return -1;
 }
};
```