Prg 8 WAP to implement functions of Dictionary using Hashing.

```
const int TS = 200;
class HashTableEntry {
        public:
    int k; int v;
HashTableEntry (int k, int v)
  { this->k = k;
    this->v = v; }
};
class HashMapTable {
  private:
    HashTableEntry **t;
  public:
    HashMapTable() {
        t = new HashTableEntry *[TS];
      for (i=0; i<TS; i++)
          t[i] = NULL;
    }
int HashFunc (int k) { return k % TS;}

void Insert (int k, int v)
  {
    int h = HashFunc(k);
    while (t[h] != NULL && t[h]->k != k)
        h = HashFunc(h+1);
```

```
        if (t[h]!=NULL)
            delete t[h];
        t[h] = new HashTableEntry (k, v);
    }

int SearchKey(int k)
  {
    int h = Hashfunc(k);
    while (t[h]!=NULL && t[h]->k!=k)
        h=Hashfunc(h+1);

    if (t[h]==NULL)
      return -1;
    else
      return t[h]->v;
  }
void Remove(int k)
  {
    int h = Hashfunc(k);
      while (t[h]!=NULL)
      {
          if(t[h]->k==k)
          break;
          h = Hashfunc(h+1);
      }
if(t[h]==NULL)
{
  cout<<"No element found at key"<<k<<endl;
  return;
```

```
}
else { delete t[h]; }
cout<<"Element Deleted" <<endl;
}
~HashMapTable()
{
    for (i=0; i<TS; i++)
    {
        if(t[i]!=NULL)
        delete t[i];
        delete[] t;
    }
};
```