PROGRAM-2

~~// Insertion~~

Arpana M Ramaswamy
1BM18CS147
04/10/2020 Arpana

Q) WAP to perform insertion, deletion & searching operation on a skip list

```
//insertion
void SkipList::InsertElement(int key)
{
  Node* curr = head;
  Node* update[MaxLevel+1];
  memset(update, 0, sizeof(Node*)*(Maxlevel+1));

for (i=level; i>=0; i--)
 {
    while (curr->forward[i] != NULL && curr->forwa
                                   -> key < key)
   curr = curr->forward[i];
   update[i] = curr;
 }

 curr = curr->forward[0];

{if (curr == NULL || curr->key != key)
```

Arpana M R
1BM18CS147
Arpana

```
r_lev = randLevel();

if(r_lev >level)
{
  for (i= level+1; i<r_lev+1; i++)
   {
    update[i] = head;
   }
  level= r_lev;
}

Node* newNode = createNode(key, r_lev);

for (i=0; i<=r_lev; i++)
  {
   newNode->forward[i] = update[i]->forward[i];
   update[i]->forward[i] = newNode;
  }

  cout<<"Entered Key "<<key<<"\n";
  }
};
```

Arpara MR
1BM18CS147
Arpara

//Deletion

```
void Skiplist::deleteElement(int key)
{
  Node *curr = head;
  Node *update[MaxLevel+1];
  memset(update, 0, sizeof(Node*)*(MaxLevel+1));

  for(i=level; i>=0; i--)
  {
   While (curr->forward[i]!=NULL && curr->forward[i]
                                     ->key<key)
       curr = curr->forward[i];
   update[i] = curr;
  }

  curr = curr->forward[0];

  if(curr!=NULL ~~&&~~ curr->key==key)
  {
   for(i=0; i<level; i++)
   {
     if (update[i]->forward[i] != curr)
       break;
    update[i]->forward[i]=current->forward[i];
   }
while (level>0 && head->forward[level]==0)
    level--;
cout<< "Deleted key is "<< key<<"\n";
}
};


// Searching

void SkipList::searchElement(int key)
{
 Node* curr = head;

 for (i= level ; i>=0; i--)
 {
     while(curr->forward[i] && curr->forward[i]->key
                                          < key)
       curr = curr->forward[i];
 }

    curr = curr->forward[0];

  if( curr && curr->key ==key)
     cout<< "Key Found "<<"\n";
  }
};
```