Skip List Program

```
#include <bits/stdc++.h>
using namespace std;

class Node
{
public:
    int key;
    Node **forward;
    Node(int, int);
};

Node::Node(int key, int level)
{
    this->key = key;
    forward = new Node*[level+1];
    memset(forward, 0,
sizeof(Node*)*(level+1));
};


class SkipList
{
```

```
    int MAXLVL;
    float P;
    int level;
    Node *header;
public:
    SkipList(int, float);
    int randomLevel();
    Node* createNode(int, int);
    void insertElement(int);
    void displayList();
};

SkipList::SkipList(int MAXLVL, float P)
{
    this->MAXLVL = MAXLVL;
    this->P = P;
    level = 0;
    header = new Node(-1, MAXLVL);
};

int SkipList::randomLevel()
{
    float r = (float)rand()/RAND_MAX;
    int lvl = 0;
    while (r < P && lvl < MAXLVL)
```

```
    {
        lvl++;
        r = (float)rand()/RAND_MAX;
    }
    return lvl;
};

Node* SkipList::createNode(int key, int level)
{
    Node *n = new Node(key, level);
    return n;
};

void SkipList::insertElement(int key)
{
    Node *current = header;
    Node *update[MAXLVL+1];
    memset(update, 0,
sizeof(Node*)*(MAXLVL+1));
    for (int i = level; i >= 0; i--)
    {
        while (current->forward[i] != NULL &&
              current->forward[i]->key < key)
            current = current->forward[i];
        update[i] = current;
    }
```

```
    }
    current = current->forward[0];

    if (current == NULL || current->key != key)
    {
        int rlevel = randomLevel();

        if (rlevel > level)
        {
            for (int i=level+1;i<rlevel+1;i++)
                update[i] = header;
            level = rlevel;
        }

        Node* n = createNode(key, rlevel);
        for (int i=0;i<=rlevel;i++)
        {
            n->forward[i] = update[i]->forward[i];
            update[i]->forward[i] = n;
        }
        cout << "Successfully Inserted key " <<
key << "\n";
    }
};
```

```
void SkipList::displayList()
{
    cout<<"\n*****Skip List*****"<<"\n";
    for (int i=level;i>=0;i--)
    {
        Node *node = header->forward[i];
        cout << "Level " << i << ": ";
        while (node != NULL)
        {
            cout << node->key<<" ";
            node = node->forward[i];
        }
        cout << "\n";
    }
};

int main()
{
    srand((unsigned)time(0));

    SkipList lst(3, 0.5);

    lst.insertElement(3);
    lst.insertElement(6);
    lst.insertElement(7);
    lst.insertElement(9);
```

```
    lst.insertElement(9);
    lst.insertElement(12);
    lst.insertElement(19);
    lst.insertElement(17);
    lst.insertElement(26);
    lst.insertElement(21);
    lst.insertElement(25);
    lst.displayList();
}
```

Output:

```
Successfully Inserted key 3
Successfully Inserted key 6
Successfully Inserted key 7
Successfully Inserted key 9
Successfully Inserted key 12
Successfully Inserted key 19
Successfully Inserted key 17
Successfully Inserted key 26
Successfully Inserted key 21
Successfully Inserted key 25

*****Skip List*****
Level 3: 3
Level 2: 3 12 21
Level 1: 3 6 12 19 21 26
Level 0: 3 6 7 9 12 17 19 21 25 26


...Program finished with exit code 0
Press ENTER to exit console.
```