Harshit Hiremath
IBM18CS036
25/11/20

ADS Lab 9

Implement all functions of dictionary using hashing.

```
struct list {
    int data;
    struct list *next;
}

class Dictionary{
    ~~node type~~ list *ptr[max], *root[max], *temp[max];
    public:
        int index;
        Dictionary() {
            index = -1;
            for (int i = 0; i<max; i++) {
                root[i] = NULL;
                ptr[i] = NULL;
                temp[i] = NULL;
            }
        }
        void insert (int key) {
            index = int( key %max);
            ptr[index] = (list *) malloc (sizeof(list));
            ptr[index]->data = key;
            if (root[index] == NULL) {
                root[index] = ptr[index]
                root[index]->next = NULL;
                temp[index] = ptr[index];
            } else {
                temp[index] = root[index];
                while ( temp[index]->next != NULL)
                    temp[index] = temp[index]->next;
                temp[index]->next = ptr[index];
            }
    }
```

```
Void search (int key){
    int flag =0;
    index = int(key % max)
    temp[index] = root[index];
    while (temp[index] != NULL){
        if ( temp[index]->data == key){
            cout << "Key found!"; << endl;
            flag = 1;
            break;
        } else temp[index] = temp[index]->next;
    }
    if (flag==0) cout << " Not found! " <<end;
}
Void delete_ele (int key) {
    index = int(key % max);
    temp [index] = root [index];
    while ( temp[index]->data != key && temp[index] != NULL
        ptr[index] = temp[index];
        temp[index] = temp[index]->next;
    }
    ptr[index] ->next = temp[index]-> next;
    cout << temp[index]->data << " has been deleted. ";
    temp[index] ->data = -1;
    free( temp[index]);
    temp[index] = NULL;
    }
}
```