

Sumanth.K.V

IBML8LS112

SC

25/10/2020

```

struct list {
    int data;
    struct list *next;

```

};

typedef struct list \* node-type;

```

node node-type ptr[10], root[10], temp[10];

```

class Dictionary

{

public:

int index;

Dictionary();

void insert(int);

void search(int);

void delete ele(int);

};

Dictionary:: Dictionary() {

index = -1;

for (i = 0; i &lt; 10; i++)

root[i] = NULL;

ptr[i] = NULL;

temp[i] = NULL;

}

(1) Sumanth.K.V

```
void Dictionary :: insert (int Key) {
```

```
    index = int (Key % 10);
```

```
    ptr[index] = (node type) malloc (Size of (node type));
```

```
    ptr[index] → data = Key;
```

```
    if (root[index] == NULL) {
```

```
        ptr[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 Dictionary :: search (int Key) {
```

```
    int flag = 0;
```

```
    index = int (Key % 10max);
```

```
    temp[index] = root[index];
```

```
    while (temp[index] != NULL) {
```

```
        if (temp[index] → data == Key) {
```

```
            cout << "Found" << endl;
```

```
            flag = 1;
```

```
            break;
```

```
        }
```

```
    else
```

```
        temp[index] = temp[index] → next;
```

```
    }
```

```
}
```

```
void Dictionary::delete_ele (int ele) {  
    index = int (ele % 10);  
    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 << "Deleted" << endl;  
    temp [index] → data = -1;  
    temp [index] = NULL;  
    free (temp [index]);  
}
```

③ dvmanth.kv