

Prog-8 : Impliment functions of directory using Hashing.

Class dictionary

{

Public :

int index;

Dictionary ();

void insert (int);

void search (int);

void delete ele (int);

};

Dictionary :: Dictionary ()

{

index = -1;

for (int i = 0 ; i < max ; i++)

{

root[i] = NULL;

ptr[i] = NULL;

temp[i] = NULL;

}

void Dictionary :: insert (int key)

{

index = int (key % max);

ptr [index] = (node-type *) malloc (sizeof (node-type));

ptr [index] -> data = key;

if (root [index] == NULL)

{

root [index] = ptr [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 % max);
```

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

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

```
    {
```

```
        if (temp [index] -> data == key)
```

```
        {
```

```
            cout << "\n Search key is found! ";
```

```
            flag = 1;
```

```
            break;
```

```
        }
```

```
        else temp [index] = temp [index] -> next;
```

```
    }
```

```
    if (flag == 0)
```

```
        cout << "\n Search key not found ";
```

```
}
```

```
void Dictionary::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 << "\n" << temp [index] -> data << " has been deleted. ";
```

```
    temp [index] -> data = -1;
```

```
    temp [index] = NULL;
```

```
    free (temp [index]);
```

```
}
```



REDMI NOTE 5 PRO
MI DUAL CAMERA