University of Asia Pacific

Department of Computer Science & Engineering

Compiler Design Lab
CSE 430

Submitted to:

Baivab Das
Lecturer
CSE, University of Asia Pacific

Submitted by:

Md. Farhad
20101073
Section: B1

Problem

Construct a simple hash-based symbol table (data dictionary) based on Chaining.

Description of the problem

We will build a symbol table in this assignment. At this initial stage, we will omit many details regarding an actual symbol table and simply add here to the basic concept that “a symbol table is an efficient data dictionary for the symbols used in a program". Thus, our assignment focuses on constructing a simple hash-based data dictionary based on chaining.

Description of the exercise code

I store information about the occurrence of various entities such as variable names, function names, objects, classes, interfaces, etc.
Here, I take the user input of a sequence of six tuples/attributes. They are

- Name
- Type
- Size
- Dimension
- Line of Code
- Address

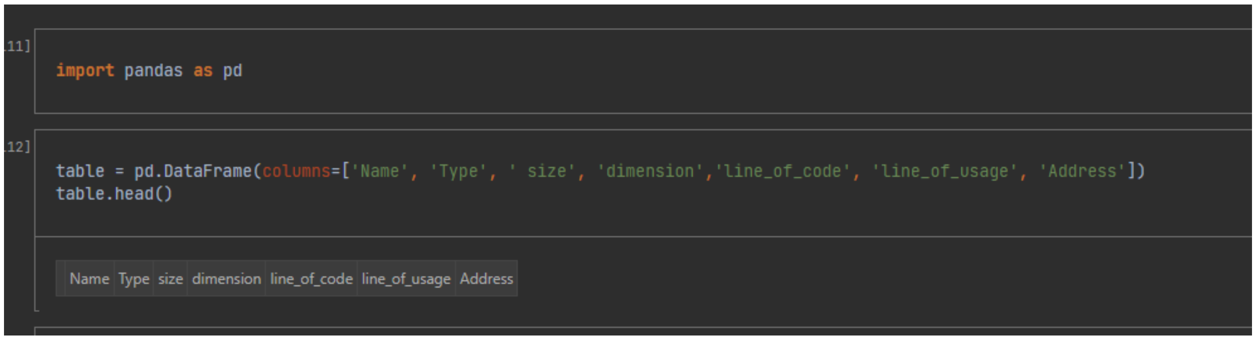
for implementation add these global functions:

(a) insert()
(b) search()
(c) delete()
(d) show()
(e) update()
(f) getHashKey()

Code & Observed output

```
import pandas as pd
table = pd.DataFrame(columns=['Name', 'Type', ' size',
'dimension','line_of_code', 'line_of_usage', 'Address'])
table.head()
```

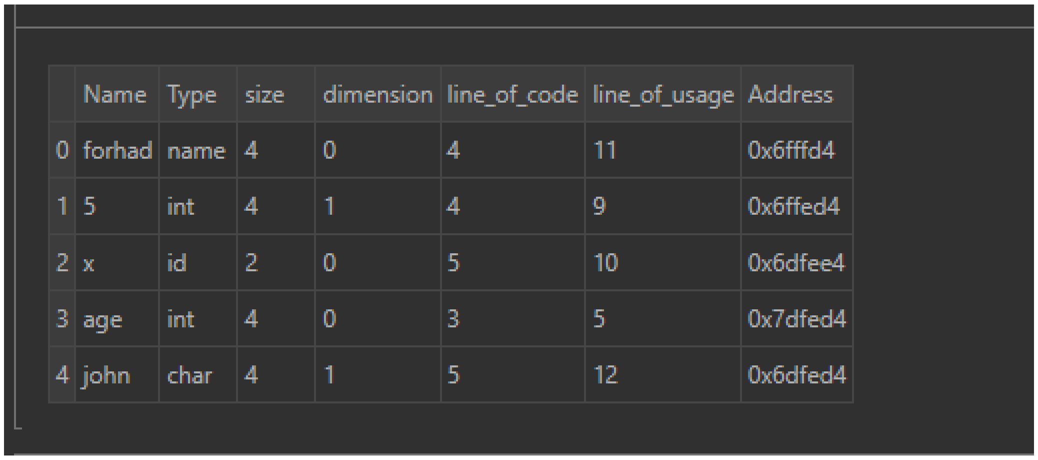
//symbolic table showing this output-



```
11]
import pandas as pd

12]
table = pd.DataFrame(columns=['Name', 'Type', ' size', 'dimension','line_of_code', 'line_of_usage', 'Address'])
table.head()

Name Type size dimension line_of_code line_of_usage Address
```

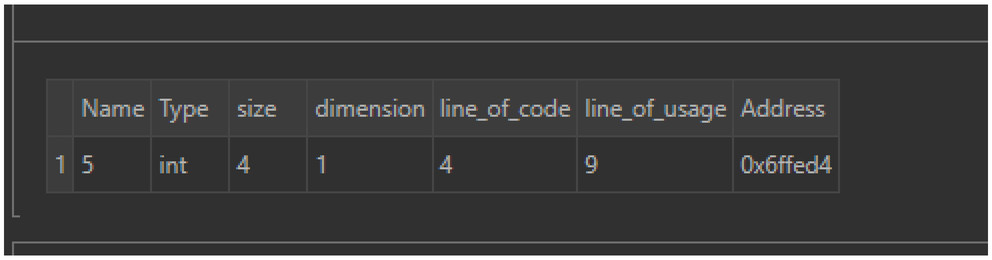
```
def insert(table, name, typ, si, di, lico, lius, add ):
    if (name not in table.Name.values) or (typ not in table.Type.values):
        table.loc[-1] = [name, typ,si, di, lico, lius, add]
        table.index = table.index + 1
        table = table.sort_index()
    return table


def search(table, name):
    return table.loc[table['Name'] == name]


def delete(table, name):
    try:
        idx = table.loc[table['Name'] == name].index[0]
        table.drop([idx], axis=0, inplace=True)
        table.reset_index(inplace=True)
        table.drop('index', axis=1, inplace=True)
        return table
    except:
        print('Name not found')
        return table
```

```
def update(table, name=None, new_name=None, typ=None, new_typ=None):
    if name:
        if name in table.Name.values:
            table.loc[table.Name == name, 'Name'] = new_name
    if typ:
        if typ in table.Type.values:
            table.loc[table.Type == typ, 'Type'] = new_typ

    return table


def show():
    print(table)


def getHashKey(table, name):
    return table.loc[table['Name'] == name].index[0]



//insert data
table = insert(table,'john', 'char', '4','1','5','12', '0x6dfed4')
table = insert(table,'age', 'int', '4','0','3','5', '0x7dfed4')
table = insert(table,'x', 'id', '2','0','5','10', '0x6dfee4')
table = insert(table,'5', 'int', '4','1','4','9', '0x6ffed4')
table = insert(table,'forhad', 'name', '4','0','4','11', '0x6fffd4')
table

//insert data show
```



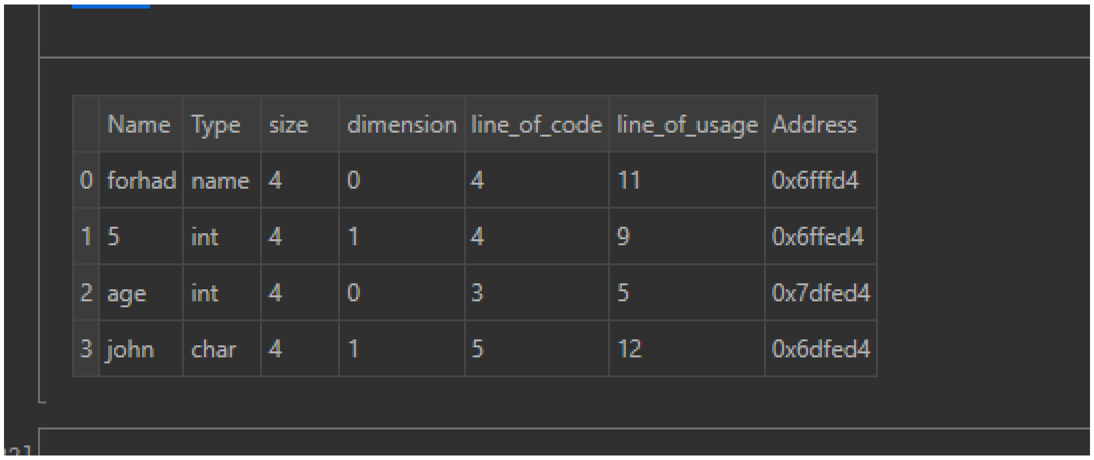
	Name	Type	size	dimension	line_of_code	line_of_usage	Address
0	forhad	name	4	0	4	11	0x6fffd4
1	5	int	4	1	4	9	0x6ffed4
2	x	id	2	0	5	10	0x6dfee4
3	age	int	4	0	3	5	0x7dfed4
4	john	char	4	1	5	12	0x6dfed4

```
//search data
search(table, '5')
```



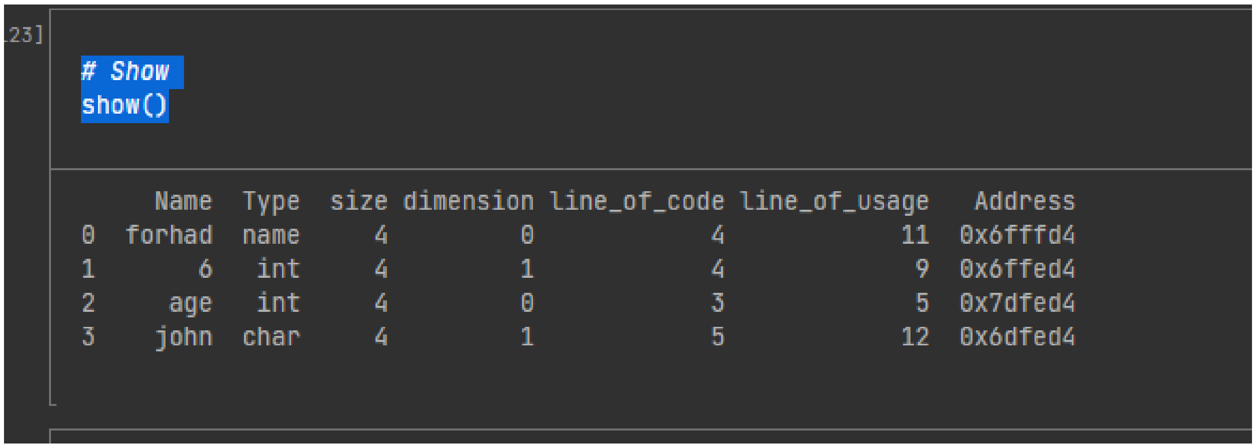
	Name	Type	size	dimension	line_of_code	line_of_usage	Address
1	5	int	4	1	4	9	0x6ffed4

```
//delete any data
table = delete(table,name='x')
table
```



	Name	Type	size	dimension	line_of_code	line_of_usage	Address
0	forhad	name	4	0	4	11	0x6fffd4
1	5	int	4	1	4	9	0x6ffed4
2	age	int	4	0	3	5	0x7dfed4
3	john	char	4	1	5	12	0x6dfed4

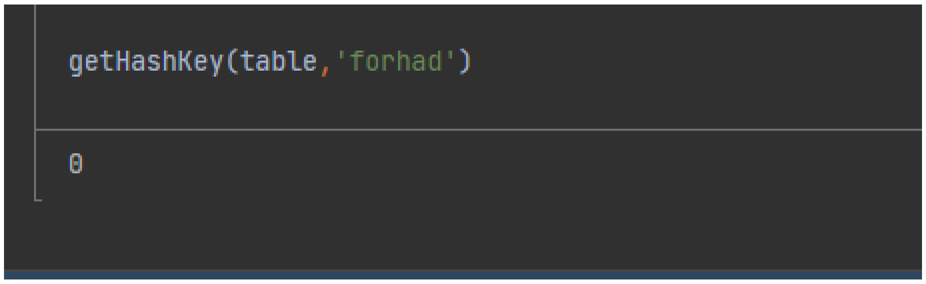
```
//update any data
table = update(table, '5','6','NUM','Number')
# Show
show()
```



```
# Show
show()
```

```
     Name  Type  size  dimension  line_of_code  line_of_usage   Address
0  forhad  name     4          0             4             11  0x6fffd4
1       6   int     4          1             4              9  0x6ffed4
2     age   int     4          0             3              5  0x7dfed4
3    john  char     4          1             5             12  0x6dfed4
```

```
//hash key

getHashKey(table,'forhad')
```



```
getHashKey(table,'forhad')
```

```
0
```