

## Assignment 7

Title: Symbol Table Hashing

### Problem Statement:

It is generated by compiler. From this perspective a symbol table is a set of name-attribute pairs. Perform the following operations on symbol table:

- Determine if particular name is in symbol table.
- 2) Retrieve the attributes of that name.
- 3) Modify the attributes of that name.
- a) Insert a new name and it's ottributes.
- ottributes. 5) Delete a name & it's attributes.

## Objective:

To understand the concept of chaining used in Hash tables.

#### Outcome:

It will be able to implement symbol table structure using hashing 4 chaining with 4 without replacement.

# Requirements:

264 bit machine operating system.

2) Editor, Compiler (g++)

3) CPU, RAM

	Theory:				
	In linear probing technique, if a				
	large number of keys are being				
	mapped to the same location				
	then searching time complexity				
	no longer remains o(1) but it can				
	fake O(n) time for search. Chaining				
	technique tries to reduce time. In				
	chaining all keys which get the				
	some location on applying hash				
	function are stored in chain.				
	The state of the s				
•	Pseudo Code:				
	1. Van surface south is from the				
•	add (symbol a)				
	if (replac)				
	if hash (S(b).k!=b) and s[b] non Empty				
	Swap (SLb], a)				
/ /	ek if s(b) non empty				
	For C=b until C=-1				
	c = SCc].chain L: First empty loc after b				
	if (no loc.)				
	error				
	else				
3 1	s(l) := a				
	s(c).chain:=l				
	1/200 · 1				
	formation of the state of the s				
	Port of the second seco				

	·
	else from l:=b
	while (sCl) not emply { hash s[n]k!=b){
	if (all loc. non empty)
	print'error' & return
	}
	if s [l] = empty
	S[l]=a ret
	if c:= l until s(c). chain =-1
	C:= S[c]. chain
	m = next empty after c
	if (not found)
	print 'error', return
	else
	S[m]:= a
	S [c]. Chain := m
	· · · · · · · · · · · · · · · · · · ·
•	retrieve (symbol a)
	b:= find (a,k)
	if (b=-1)
	print "not found"
	else
	Read attr. of s(b)
	print attr.
	return.
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

200		•
update	(symbol	a)

return.

print (not found) ret

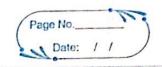
if (loc = found) SCLJ. chain = SCbJ. chain

make S(b) = empty

return.

## \* Problems. faced

- 1. Probing without replacement 2. Collision resolution.



•	Test cases:			
	Input	Operation	Output	Result
· · · · · · · · · · · · · · · · · · ·	Bde, Abed, Def	find (Abc)	loc: 0	Success
	(without Replacement)	find (Bcd)	loc:2	
		Find (Abcd)	loc: 4	
		delefe (Ade)	deleted	Ţ <sup>i</sup>
P	Conclusion:			
	hashing usin			
	without replace	egvera:		