**LINEAR PROBING:**

1. #include<iostream>
2. #include<cstdlib>
3. using namespace std;
4. struct student
5. {
6. int rno;
7. char name[20];
8. student \*link;
9. }\*head,\*temp,\*temp1,\*p;
10. struct node
11. {
12. int key;
13. student \*slink;
14. }\*t,\*prev;
15. node \*hash[20];
16. int n=7;
17. void del1(student \*n)
18. {
19. temp=head;
20. while(temp->rno!=n->rno)
21. {
22. p=temp;
23. temp=temp->link;
24. }
25. if(temp==head)
26. head=head->link;
27. else if(temp->link=='\0')
28. p->link='\0';
29. else
30. {
31. p->link=temp->link;
32. temp->link='\0';
33. }
34. }
35. void del()
36. {
37. int no,flag=0;
38. cout<<"\n enter roll no to be deleted:";
39. cin>>no;
40. student \*n=new student;
41. for(int i=0;i<7;++i)
42. {
43. if(hash[i]->key==no)
44. {
45. flag=1;
46. cout<<"\n student deleted:";
47. n=hash[i]->slink;
48. hash[i]='\0';
49. del1(n);
50. break;
51. }
52. }
53. if(flag==0)
54. cout<<"\n student not found";
55. }
56. void insert()
57. {
58. cout<<"\n enter student details:";
59. student \*s=new student;
60. s->link='\0';
61. cout<<"\n enter rno:";
62. cin>>s->rno;
63. cout<<"\n enter name:";
64. cin>>s->name;
65. int k;
66. k=(s->rno)%7;
67. node \*n=new node;
68. n->key=s->rno;
69. n->slink=s;
70. int x=0;
71. for(int i=0;i<7;++i)
72. {
73. if(k+i >6)
74. {
75. k=(k+i)%7;
76. i=0;
77. }
78. if(hash[k+i]=='\0')
79. {
80. hash[k+i]=n;
81. if(head=='\0')
82. {
83. head=s;
84. temp1=s;
85. }
86. else
87. {
88. temp1->link=s;
89. temp1=s;
90. }
91. cout<<"\n inserted into the list:";
92. cout<<"\n inserted into hash table:";
93. break;
94. }
95. if(x==6)
96. cout<<"\n no more insertions";
97. ++x;
98. }
99. }
100. void displayhash()
101. {
102. for(int i=0;i<7;++i)
103. {
104. cout<<i<<"->";
105. if(hash[i]!='\0')
106. cout<<hash[i]->key;
107. cout<<"\n";
108. }
109. }
110. void displaylist()
111. {
112. temp=head;
113. while(temp!='\0')
114. {
115. cout<<temp->rno<<" "<<temp->name;
116. temp=temp->link;
117. cout<<"\n";
118. }
119. }
120. void search()
121. {
122. int key,flag=0;
123. cout<<"\n enter the rno to be searched:";
124. cin>>key;
125. student \*n=new student;
126. for(int i=0;i<7;++i)
127. {
128. if(hash[i]->key==key)
129. {
130. flag=1;
131. n=hash[i]->slink;
132. break;
133. }
134. }
135. if(flag==0)
136. cout<<"\n student not found";
137. else
138. {
139. cout<<"\n Roll no:"<<n->rno;
140. cout<<"\n Name:"<<n->name;
141. }
142. }
143. int main()
144. {
145. int ch;
146. do
147. {
148. cout<<"\n1.insert \n2.delete \n3.display list \n4.display hash table \n5.search \n6.exit:";
149. cin>>ch;
150. switch(ch)
151. {
152. case 1:insert();
153. break;
154. case 2:del();
155. break;
156. case 3:displaylist();
157. break;
158. case 4:displayhash();
159. break;
160. case 5:search();
161. break;
162. case 6:exit(0);
163. }
164. }while(ch!=6);
165. return 0;
166. }