**QUADRATIC 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 y;
71. if(hash[k]=='\0')
72. {
73. hash[k]=n;
74. if(head=='\0')
75. {
76. head=s;
77. temp1=s;
78. }
79. else
80. {
81. temp1->link=s;
82. temp1=s;
83. }
84. cout<<"\n inserted into the list:";
85. cout<<"\n inserted into hash table:";
86. }
87. else
88. {
89. y=k;
90. for(int i=0;i<7;++i)
91. {
92. k=y;
93. k=(k+(i\*i))%7;
94. if(hash[k]=='\0')
95. {
96. hash[k]=n;
97. if(head=='\0')
98. {
99. head=s;
100. temp1=s;
101. }
102. else
103. {
104. temp1->link=s;
105. temp1=s;
106. }
107. cout<<"\n inserted into the list:";
108. cout<<"\n inserted into hash table:";
109. break;
110. }
111. }}
112. }
113. void displayhash()
114. {
115. for(int i=0;i<7;++i)
116. {
117. cout<<i<<"->";
118. if(hash[i]!='\0')
119. cout<<hash[i]->key;
120. cout<<"\n";
121. }
122. }
123. void displaylist()
124. {
125. temp=head;
126. while(temp!='\0')
127. {
128. cout<<temp->rno<<" "<<temp->name;
129. temp=temp->link;
130. cout<<"\n";
131. }
132. }
133. void search()
134. {
135. int key,flag=0;
136. cout<<"\n enter the rno to be searched:";
137. cin>>key;
138. student \*n=new student;
139. for(int i=0;i<7;++i)
140. {
141. if(hash[i]->key==key)
142. {
143. flag=1;
144. n=hash[i]->slink;
145. break;
146. }
147. }
148. if(flag==0)
149. cout<<"\n student not found";
150. else
151. {
152. cout<<"\n Roll no:"<<n->rno;
153. cout<<"\n Name:"<<n->name;
154. }
155. }
156. int main()
157. {
158. int ch,x;
159. do
160. {
161. cout<<"\n1.insert \n2.delete \n3.display list \n4.display hash table \n5.search \n6.exit:";
162. cin>>ch;
163. switch(ch)
164. {
165. case 1:x=0;
     1. for(int i=0;i<7;++i)
166. {
167. if(hash[i]!='\0')
168. ++x;}
169. if(x==7)
170. {cout<<"\n no more insertions";
171. break;
172. }
173. insert();
174. break;
175. case 2:del();
176. break;
177. case 3:displaylist();
178. break;
179. case 4:displayhash();
180. break;
181. case 5:search();
182. break;
183. case 6:exit(0);
184. }
185. }while(ch!=6);
186. return 0;
187. }