

Laboratorio #6: Hash

Nombre: Kaiser Obaldía

Cédula: 8-898-703

Para este laboratorio, se le pide implementar una clase que represente una tabla de símbolos mediante una tabla hash (la clave de la tabla hash es el símbolo de una tabla de símbolos). Esta es una estructura donde el símbolo o clave puede ser numérica o un cadena de caracteres, de manera que al definir un objeto de la clase se instancie a un tipo específico.

El valor item asociado a la clave, y que se almacena en la tabla, puede ser un valor o una estructura, lo que implica que, al igual que la clave, el tipo asociado al item es instanciado una vez definido un objeto de esta clase. Esto es similar a la clase Hashtable de C++, pero que en este caso se implementa como tabla hash.

Las operaciones básicas de esta clase son:

- agregar elementos a la tabla
- preguntar si la tabla esta vacía
- retornar la primera clave de la tabla
- retornar la última clave de la tabla
- retornar la siguiente clave a una clave dada
- retornar el item (donde item puede ser una estructura) asociado a una clave dada

La clase Hashtable

La clase Hashtable almacena pares del tipo clave/valor en una tabla desordenada. Al utilizar Hashtable se proporciona un objeto que sirve como clave y otro que sirve de valor que se desea vincular a esta clave. Un objeto de este tipo proporciona un modo rápido de inserción y de búsqueda.

Una analogía es una lista telefónica o un diccionario, dado un nombre proporcionar un número de teléfono o dada una palabra proporcionar una definición.

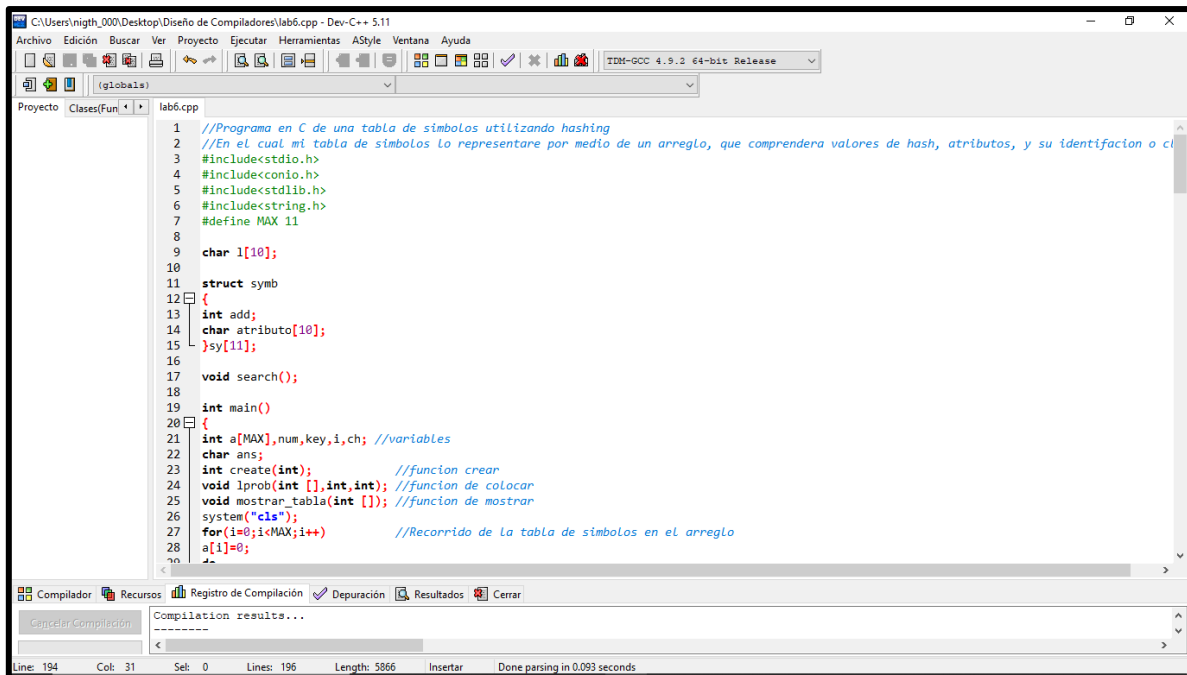
En un vector se obtienen los valores mediante un índice entero. Con una tabla desordenada no se utiliza un entero directamente sino que se utiliza lo que se denomina como código. Los códigos asignan un número a un objeto.

El código permite buscar un valor en un solo lugar. Si no se encuentra nada tras convertir la clave al código entonces la clave no se encuentra en la tabla.

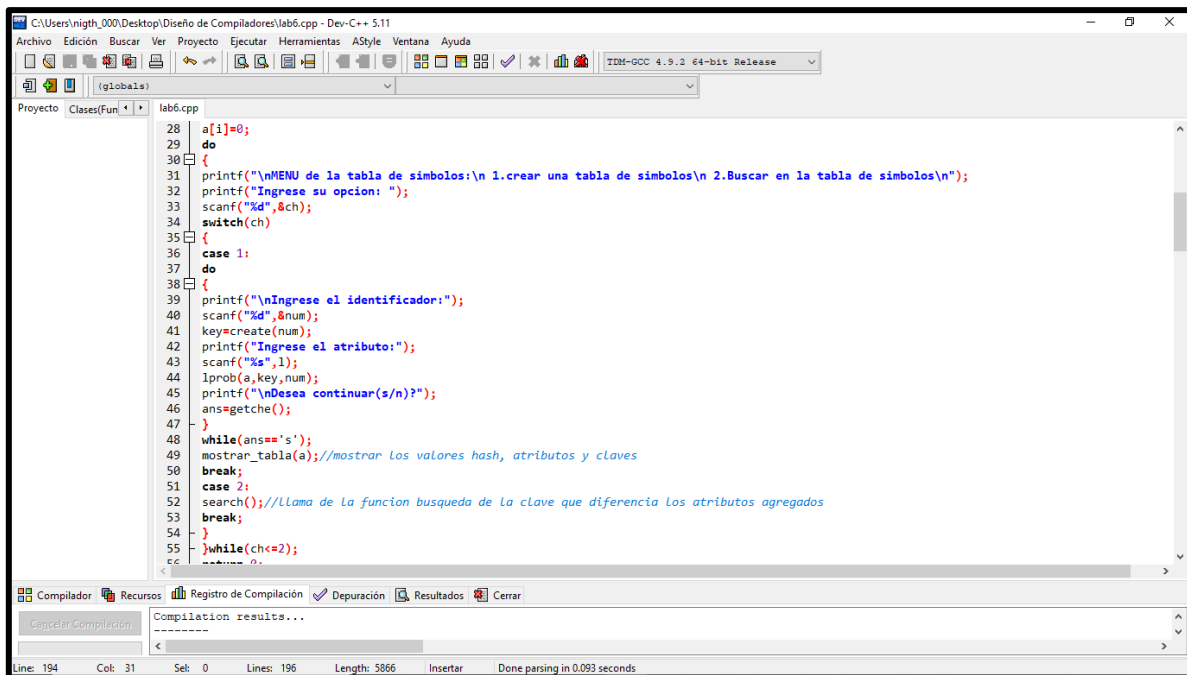
...

```
Hashtable ht = new Hashtable ( 23 );
ht.put ( "uno", new Integer ( 1 ) );
ht.put ( "dos", new Integer ( 2 ) );
ht.put ( "tres", new Integer ( 3 ) );
ht.put ( "cuatro", new Integer ( 4 ) );
System.out.println ( (Integer) ht.get ( "uno" ) );
System.out.println ( (Integer) ht.get ( "tres" ) );
System.out.println ( (Integer) ht.get ( "cinco" ) )
```

Desarrollo: capturas



```
1 //Programa en C de una tabla de simbolos utilizando hashing
2 //En el cual mi tabla de simbolos lo representare por medio de un arreglo, que comprendera valores de hash, atributos, y su identificacion o cl
3 #include<stdio.h>
4 #include<conio.h>
5 #include<stdlib.h>
6 #include<string.h>
7 #define MAX 11
8
9 char l[10];
10
11 struct symb
12 {
13     int add;
14     char atributo[10];
15 }sy[11];
16
17 void search();
18
19 int main()
20 {
21     int a[MAX], num, key, i, ch; //variables
22     char ans;
23     int create(int); //funcion crear
24     void lprob(int [],int,int); //funcion de colocar
25     void mostrar_tabla(int []); //funcion de mostrar
26     system("cls");
27     for(i=0;i<MAX;i++) //Recorrido de la tabla de simbolos en el arreglo
28     {
29         a[i]=0;
30     }
31 }
```



```
28 a[i]=0;
29 do
30 {
31     printf("\nMENU de la tabla de simbolos:\n 1.crear una tabla de simbolos\n 2.Buscar en la tabla de simbolos\n");
32     printf("Ingrese su opcion: ");
33     scanf("%d",&ch);
34     switch(ch)
35     {
36     case 1:
37     do
38     {
39         printf("\nIngrese el identificador:");
40         scanf("%d",&num);
41         key=create(num);
42         printf("Ingrese el atributo:");
43         scanf("%s",l);
44         lprob(a,key,num);
45         printf("\nDesea continuar(s/n)?");
46         ans=getche();
47     }
48     while(ans=='s');
49     mostrar_tabla(a); //mostrar Los valores hash, atributos y claves
50     break;
51     case 2:
52     search(); //Llama de la funcion busqueda de la clave que diferencia Los atributos agregados
53     break;
54     }
55 }while(ch<=2);
56 }
```

```
C:\Users\nighth_000\Desktop\Diseño de Compiladores\lab6.cpp - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
(globals)
Proyecto Clases(Funciones) lab6.cpp
56     return 0;
57 }
58
59 int create(int num)
60 {
61     int key;
62     key=num%11;
63     return key;
64 }
65
66 void lprob(int a[MAX],int key,int num)//Funcion que permite hacer el llenado de Los valores hash, atributos y clavrs
67 {
68     int flag,i,count=0;
69     void mostrar_tabla(int a[]); //Llama a La funcion de mostrar tabla y se verifica que este vacía La tabla. Si lo esta se procede el llenado.
70     flag=0;
71     if(a[key]==0)//Condición para verificar que este vacía La tabla
72     {
73         a[key]=num;
74         sy[key].add=num;
75         strcpy(sy[key].atributo,1);//copia los elementos agregados a el arreglo
76     }
77     else
78     {
79         i=0;
80         while(i<MAX)//mientras Los elementos agregados no sobrepasen el tope, continua agregando
81         {
82             if(a[i]!=0)
83                 count++;
84             i++;
85         }
86     }
87 }
88
89 int main()
90 {
91     int num;
92     int key;
93     int i;
94     int MAX=10;
95     int sy[10][10];
96     sy[0][0]=0;
97     sy[0][1]=0;
98     sy[0][2]=0;
99     sy[0][3]=0;
100     sy[0][4]=0;
101     sy[0][5]=0;
102     sy[0][6]=0;
103     sy[0][7]=0;
104     sy[0][8]=0;
105     sy[0][9]=0;
106     sy[1][0]=0;
107     sy[1][1]=0;
108     sy[1][2]=0;
109     sy[1][3]=0;
110     sy[1][4]=0;
111     sy[1][5]=0;
112     sy[1][6]=0;
113     sy[1][7]=0;
114     sy[1][8]=0;
115     sy[1][9]=0;
116     sy[2][0]=0;
117     sy[2][1]=0;
118     sy[2][2]=0;
119     sy[2][3]=0;
120     sy[2][4]=0;
121     sy[2][5]=0;
122     sy[2][6]=0;
123     sy[2][7]=0;
124     sy[2][8]=0;
125     sy[2][9]=0;
126     sy[3][0]=0;
127     sy[3][1]=0;
128     sy[3][2]=0;
129     sy[3][3]=0;
130     sy[3][4]=0;
131     sy[3][5]=0;
132     sy[3][6]=0;
133     sy[3][7]=0;
134     sy[3][8]=0;
135     sy[3][9]=0;
136     sy[4][0]=0;
137     sy[4][1]=0;
138     sy[4][2]=0;
139     sy[4][3]=0;
140     sy[4][4]=0;
141     sy[4][5]=0;
142     sy[4][6]=0;
143     sy[4][7]=0;
144     sy[4][8]=0;
145     sy[4][9]=0;
146     sy[5][0]=0;
147     sy[5][1]=0;
148     sy[5][2]=0;
149     sy[5][3]=0;
150     sy[5][4]=0;
151     sy[5][5]=0;
152     sy[5][6]=0;
153     sy[5][7]=0;
154     sy[5][8]=0;
155     sy[5][9]=0;
156     sy[6][0]=0;
157     sy[6][1]=0;
158     sy[6][2]=0;
159     sy[6][3]=0;
160     sy[6][4]=0;
161     sy[6][5]=0;
162     sy[6][6]=0;
163     sy[6][7]=0;
164     sy[6][8]=0;
165     sy[6][9]=0;
166     sy[7][0]=0;
167     sy[7][1]=0;
168     sy[7][2]=0;
169     sy[7][3]=0;
170     sy[7][4]=0;
171     sy[7][5]=0;
172     sy[7][6]=0;
173     sy[7][7]=0;
174     sy[7][8]=0;
175     sy[7][9]=0;
176     sy[8][0]=0;
177     sy[8][1]=0;
178     sy[8][2]=0;
179     sy[8][3]=0;
180     sy[8][4]=0;
181     sy[8][5]=0;
182     sy[8][6]=0;
183     sy[8][7]=0;
184     sy[8][8]=0;
185     sy[8][9]=0;
186     sy[9][0]=0;
187     sy[9][1]=0;
188     sy[9][2]=0;
189     sy[9][3]=0;
190     sy[9][4]=0;
191     sy[9][5]=0;
192     sy[9][6]=0;
193     sy[9][7]=0;
194     sy[9][8]=0;
195     sy[9][9]=0;
196     while(1)
197     {
198         num=rand()%100;
199         key=create(num);
200         lprob(sy,key,num);
201         printf("El valor de la clave es: %d\n",key);
202         printf("El valor de la tabla es: %d\n",count);
203         mostrar_tabla(sy);
204         getch();
205         exit(1);
206     }
207 }
208
209 void mostrar_tabla(int a[])
210 {
211     int i;
212     for(i=0;i<MAX;i++)
213     {
214         if(a[i]!=0)
215             printf("%d ",a[i]);
216         else
217             printf("0 ");
218     }
219     printf("\n");
220 }
221
222 void strcpy(char s1[],char s2[])
223 {
224     int i;
225     for(i=0;s2[i]!='\0';i++)
226         s1[i]=s2[i];
227     s1[i]='\0';
228 }
229
230 void add(char s1[],char s2[])
231 {
232     int i;
233     for(i=0;s2[i]!='\0';i++)
234         s1[i]=s2[i];
235     s1[i]='\0';
236 }
237
238 void atributo(char s1[],char s2[])
239 {
240     int i;
241     for(i=0;s2[i]!='\0';i++)
242         s1[i]=s2[i];
243     s1[i]='\0';
244 }
245
246 void atributo(char s1[],char s2[])
247 {
248     int i;
249     for(i=0;s2[i]!='\0';i++)
250         s1[i]=s2[i];
251     s1[i]='\0';
252 }
253
254 void atributo(char s1[],char s2[])
255 {
256     int i;
257     for(i=0;s2[i]!='\0';i++)
258         s1[i]=s2[i];
259     s1[i]='\0';
260 }
261
262 void atributo(char s1[],char s2[])
263 {
264     int i;
265     for(i=0;s2[i]!='\0';i++)
266         s1[i]=s2[i];
267     s1[i]='\0';
268 }
269
270 void atributo(char s1[],char s2[])
271 {
272     int i;
273     for(i=0;s2[i]!='\0';i++)
274         s1[i]=s2[i];
275     s1[i]='\0';
276 }
277
278 void atributo(char s1[],char s2[])
279 {
280     int i;
281     for(i=0;s2[i]!='\0';i++)
282         s1[i]=s2[i];
283     s1[i]='\0';
284 }
285
286 void atributo(char s1[],char s2[])
287 {
288     int i;
289     for(i=0;s2[i]!='\0';i++)
290         s1[i]=s2[i];
291     s1[i]='\0';
292 }
293
294 void atributo(char s1[],char s2[])
295 {
296     int i;
297     for(i=0;s2[i]!='\0';i++)
298         s1[i]=s2[i];
299     s1[i]='\0';
300 }
301
302 void atributo(char s1[],char s2[])
303 {
304     int i;
305     for(i=0;s2[i]!='\0';i++)
306         s1[i]=s2[i];
307     s1[i]='\0';
308 }
309
310 void atributo(char s1[],char s2[])
311 {
312     int i;
313     for(i=0;s2[i]!='\0';i++)
314         s1[i]=s2[i];
315     s1[i]='\0';
316 }
317
318 void atributo(char s1[],char s2[])
319 {
320     int i;
321     for(i=0;s2[i]!='\0';i++)
322         s1[i]=s2[i];
323     s1[i]='\0';
324 }
325
326 void atributo(char s1[],char s2[])
327 {
328     int i;
329     for(i=0;s2[i]!='\0';i++)
330         s1[i]=s2[i];
331     s1[i]='\0';
332 }
333
334 void atributo(char s1[],char s2[])
335 {
336     int i;
337     for(i=0;s2[i]!='\0';i++)
338         s1[i]=s2[i];
339     s1[i]='\0';
340 }
341
342 void atributo(char s1[],char s2[])
343 {
344     int i;
345     for(i=0;s2[i]!='\0';i++)
346         s1[i]=s2[i];
347     s1[i]='\0';
348 }
349
350 void atributo(char s1[],char s2[])
351 {
352     int i;
353     for(i=0;s2[i]!='\0';i++)
354         s1[i]=s2[i];
355     s1[i]='\0';
356 }
357
358 void atributo(char s1[],char s2[])
359 {
360     int i;
361     for(i=0;s2[i]!='\0';i++)
362         s1[i]=s2[i];
363     s1[i]='\0';
364 }
365
366 void atributo(char s1[],char s2[])
367 {
368     int i;
369     for(i=0;s2[i]!='\0';i++)
370         s1[i]=s2[i];
371     s1[i]='\0';
372 }
373
374 void atributo(char s1[],char s2[])
375 {
376     int i;
377     for(i=0;s2[i]!='\0';i++)
378         s1[i]=s2[i];
379     s1[i]='\0';
380 }
381
382 void atributo(char s1[],char s2[])
383 {
384     int i;
385     for(i=0;s2[i]!='\0';i++)
386         s1[i]=s2[i];
387     s1[i]='\0';
388 }
389
390 void atributo(char s1[],char s2[])
391 {
392     int i;
393     for(i=0;s2[i]!='\0';i++)
394         s1[i]=s2[i];
395     s1[i]='\0';
396 }
397
398 void atributo(char s1[],char s2[])
399 {
400     int i;
401     for(i=0;s2[i]!='\0';i++)
402         s1[i]=s2[i];
403     s1[i]='\0';
404 }
405
406 void atributo(char s1[],char s2[])
407 {
408     int i;
409     for(i=0;s2[i]!='\0';i++)
410         s1[i]=s2[i];
411     s1[i]='\0';
412 }
413
414 void atributo(char s1[],char s2[])
415 {
416     int i;
417     for(i=0;s2[i]!='\0';i++)
418         s1[i]=s2[i];
419     s1[i]='\0';
420 }
421
422 void atributo(char s1[],char s2[])
423 {
424     int i;
425     for(i=0;s2[i]!='\0';i++)
426         s1[i]=s2[i];
427     s1[i]='\0';
428 }
429
430 void atributo(char s1[],char s2[])
431 {
432     int i;
433     for(i=0;s2[i]!='\0';i++)
434         s1[i]=s2[i];
435     s1[i]='\0';
436 }
437
438 void atributo(char s1[],char s2[])
439 {
440     int i;
441     for(i=0;s2[i]!='\0';i++)
442         s1[i]=s2[i];
443     s1[i]='\0';
444 }
445
446 void atributo(char s1[],char s2[])
447 {
448     int i;
449     for(i=0;s2[i]!='\0';i++)
450         s1[i]=s2[i];
451     s1[i]='\0';
452 }
453
454 void atributo(char s1[],char s2[])
455 {
456     int i;
457     for(i=0;s2[i]!='\0';i++)
458         s1[i]=s2[i];
459     s1[i]='\0';
460 }
461
462 void atributo(char s1[],char s2[])
463 {
464     int i;
465     for(i=0;s2[i]!='\0';i++)
466         s1[i]=s2[i];
467     s1[i]='\0';
468 }
469
470 void atributo(char s1[],char s2[])
471 {
472     int i;
473     for(i=0;s2[i]!='\0';i++)
474         s1[i]=s2[i];
475     s1[i]='\0';
476 }
477
478 void atributo(char s1[],char s2[])
479 {
480     int i;
481     for(i=0;s2[i]!='\0';i++)
482         s1[i]=s2[i];
483     s1[i]='\0';
484 }
485
486 void atributo(char s1[],char s2[])
487 {
488     int i;
489     for(i=0;s2[i]!='\0';i++)
490         s1[i]=s2[i];
491     s1[i]='\0';
492 }
493
494 void atributo(char s1[],char s2[])
495 {
496     int i;
497     for(i=0;s2[i]!='\0';i++)
498         s1[i]=s2[i];
499     s1[i]='\0';
500 }
501
502 void atributo(char s1[],char s2[])
503 {
504     int i;
505     for(i=0;s2[i]!='\0';i++)
506         s1[i]=s2[i];
507     s1[i]='\0';
508 }
509
510 void atributo(char s1[],char s2[])
511 {
512     int i;
513     for(i=0;s2[i]!='\0';i++)
514         s1[i]=s2[i];
515     s1[i]='\0';
516 }
517
518 void atributo(char s1[],char s2[])
519 {
520     int i;
521     for(i=0;s2[i]!='\0';i++)
522         s1[i]=s2[i];
523     s1[i]='\0';
524 }
525
526 void atributo(char s1[],char s2[])
527 {
528     int i;
529     for(i=0;s2[i]!='\0';i++)
530         s1[i]=s2[i];
531     s1[i]='\0';
532 }
533
534 void atributo(char s1[],char s2[])
535 {
536     int i;
537     for(i=0;s2[i]!='\0';i++)
538         s1[i]=s2[i];
539     s1[i]='\0';
540 }
541
542 void atributo(char s1[],char s2[])
543 {
544     int i;
545     for(i=0;s2[i]!='\0';i++)
546         s1[i]=s2[i];
547     s1[i]='\0';
548 }
549
550 void atributo(char s1[],char s2[])
551 {
552     int i;
553     for(i=0;s2[i]!='\0';i++)
554         s1[i]=s2[i];
555     s1[i]='\0';
556 }
557
558 void atributo(char s1[],char s2[])
559 {
560     int i;
561     for(i=0;s2[i]!='\0';i++)
562         s1[i]=s2[i];
563     s1[i]='\0';
564 }
565
566 void atributo(char s1[],char s2[])
567 {
568     int i;
569     for(i=0;s2[i]!='\0';i++)
570         s1[i]=s2[i];
571     s1[i]='\0';
572 }
573
574 void atributo(char s1[],char s2[])
575 {
576     int i;
577     for(i=0;s2[i]!='\0';i++)
578         s1[i]=s2[i];
579     s1[i]='\0';
580 }
581
582 void atributo(char s1[],char s2[])
583 {
584     int i;
585     for(i=0;s2[i]!='\0';i++)
586         s1[i]=s2[i];
587     s1[i]='\0';
588 }
589
590 void atributo(char s1[],char s2[])
591 {
592     int i;
593     for(i=0;s2[i]!='\0';i++)
594         s1[i]=s2[i];
595     s1[i]='\0';
596 }
597
598 void atributo(char s1[],char s2[])
599 {
600     int i;
601     for(i=0;s2[i]!='\0';i++)
602         s1[i]=s2[i];
603     s1[i]='\0';
604 }
605
606 void atributo(char s1[],char s2[])
607 {
608     int i;
609     for(i=0;s2[i]!='\0';i++)
610         s1[i]=s2[i];
611     s1[i]='\0';
612 }
613
614 void atributo(char s1[],char s2[])
615 {
616     int i;
617     for(i=0;s2[i]!='\0';i++)
618         s1[i]=s2[i];
619     s1[i]='\0';
620 }
621
622 void atributo(char s1[],char s2[])
623 {
624     int i;
625     for(i=0;s2[i]!='\0';i++)
626         s1[i]=s2[i];
627     s1[i]='\0';
628 }
629
630 void atributo(char s1[],char s2[])
631 {
632     int i;
633     for(i=0;s2[i]!='\0';i++)
634         s1[i]=s2[i];
635     s1[i]='\0';
636 }
637
638 void atributo(char s1[],char s2[])
639 {
640     int i;
641     for(i=0;s2[i]!='\0';i++)
642         s1[i]=s2[i];
643     s1[i]='\0';
644 }
645
646 void atributo(char s1[],char s2[])
647 {
648     int i;
649     for(i=0;s2[i]!='\0';i++)
650         s1[i]=s2[i];
651     s1[i]='\0';
652 }
653
654 void atributo(char s1[],char s2[])
655 {
656     int i;
657     for(i=0;s2[i]!='\0';i++)
658         s1[i]=s2[i];
659     s1[i]='\0';
660 }
661
662 void atributo(char s1[],char s2[])
663 {
664     int i;
665     for(i=0;s2[i]!='\0';i++)
666         s1[i]=s2[i];
667     s1[i]='\0';
668 }
669
670 void atributo(char s1[],char s2[])
671 {
672     int i;
673     for(i=0;s2[i]!='\0';i++)
674         s1[i]=s2[i];
675     s1[i]='\0';
676 }
677
678 void atributo(char s1[],char s2[])
679 {
680     int i;
681     for(i=0;s2[i]!='\0';i++)
682         s1[i]=s2[i];
683     s1[i]='\0';
684 }
685
686 void atributo(char s1[],char s2[])
687 {
688     int i;
689     for(i=0;s2[i]!='\0';i++)
690         s1[i]=s2[i];
691     s1[i]='\0';
692 }
693
694 void atributo(char s1[],char s2[])
695 {
696     int i;
697     for(i=0;s2[i]!='\0';i++)
698         s1[i]=s2[i];
699     s1[i]='\0';
700 }
701
702 void atributo(char s1[],char s2[])
703 {
704     int i;
705     for(i=0;s2[i]!='\0';i++)
706         s1[i]=s2[i];
707     s1[i]='\0';
708 }
709
710 void atributo(char s1[],char s2[])
711 {
712     int i;
713     for(i=0;s2[i]!='\0';i++)
714         s1[i]=s2[i];
715     s1[i]='\0';
716 }
717
718 void atributo(char s1[],char s2[])
719 {
720     int i;
721     for(i=0;s2[i]!='\0';i++)
722         s1[i]=s2[i];
723     s1[i]='\0';
724 }
725
726 void atributo(char s1[],char s2[])
727 {
728     int i;
729     for(i=0;s2[i]!='\0';i++)
730         s1[i]=s2[i];
731     s1[i]='\0';
732 }
733
734 void atributo(char s1[],char s2[])
735 {
736     int i;
737     for(i=0;s2[i]!='\0';i++)
738         s1[i]=s2[i];
739     s1[i]='\0';
740 }
741
742 void atributo(char s1[],char s2[])
743 {
744     int i;
745     for(i=0;s2[i]!='\0';i++)
746         s1[i]=s2[i];
747     s1[i]='\0';
748 }
749
750 void atributo(char s1[],char s2[])
751 {
752     int i;
753     for(i=0;s2[i]!='\0';i++)
754         s1[i]=s2[i];
755     s1[i]='\0';
756 }
757
758 void atributo(char s1[],char s2[])
759 {
760     int i;
761     for(i=0;s2[i]!='\0';i++)
762         s1[i]=s2[i];
763     s1[i]='\0';
764 }
765
766 void atributo(char s1[],char s2[])
767 {
768     int i;
769     for(i=0;s2[i]!='\0';i++)
770         s1[i]=s2[i];
771     s1[i]='\0';
772 }
773
774 void atributo(char s1[],char s2[])
775 {
776     int i;
777     for(i=0;s2[i]!='\0';i++)
778         s1[i]=s2[i];
779     s1[i]='\0';
780 }
781
782 void atributo(char s1[],char s2[])
783 {
784     int i;
785     for(i=0;s2[i]!='\0';i++)
786         s1[i]=s2[i];
787     s1[i]='\0';
788 }
789
790 void atributo(char s1[],char s2[])
791 {
792     int i;
793     for(i=0;s2[i]!='\0';i++)
794         s1[i]=s2[i];
795     s1[i]='\0';
796 }
797
798 void atributo(char s1[],char s2[])
799 {
800     int i;
801     for(i=0;s2[i]!='\0';i++)
802         s1[i]=s2[i];
803     s1[i]='\0';
804 }
805
806 void atributo(char s1[],char s2[])
807 {
808     int i;
809     for(i=0;s2[i]!='\0';i++)
810         s1[i]=s2[i];
811     s1[i]='\0';
812 }
813
814 void atributo(char s1[],char s2[])
815 {
816     int i;
817     for(i=0;s2[i]!='\0';i++)
818         s1[i]=s2[i];
819     s1[i]='\0';
820 }
821
822 void atributo(char s1[],char s2[])
823 {
824     int i;
825     for(i=0;s2[i]!='\0';i++)
826         s1[i]=s2[i];
827     s1[i]='\0';
828 }
829
830 void atributo(char s1[],char s2[])
831 {
832     int i;
833     for(i=0;s2[i]!='\0';i++)
834         s1[i]=s2[i];
835     s1[i]='\0';
836 }
837
838 void atributo(char s1[],char s2[])
839 {
840     int i;
841     for(i=0;s2[i]!='\0';i++)
842         s1[i]=s2[i];
843     s1[i]='\0';
844 }
845
846 void atributo(char s1[],char s2[])
847 {
848     int i;
849     for(i=0;s2[i]!='\0';i++)
850         s1[i]=s2[i];
851     s1[i]='\0';
852 }
853
854 void atributo(char s1[],char s2[])
855 {
856     int i;
857     for(i=0;s2[i]!='\0';i++)
858         s1[i]=s2[i];
859     s1[i]='\0';
860 }
861
862 void atributo(char s1[],char s2[])
863 {
864     int i;
865     for(i=0;s2[i]!='\0';i++)
866         s1[i]=s2[i];
867     s1[i]='\0';
868 }
869
870 void atributo(char s1[],char s2[])
871 {
872     int i;
873     for(i=0;s2[i]!='\0';i++)
874         s1[i]=s2[i];
875     s1[i]='\0';
876 }
877
878 void atributo(char s1[],char s2[])
879 {
880     int i;
881     for(i=0;s2[i]!='\0';i++)
882         s1[i]=s2[i];
883     s1[i]='\0';
884 }
885
886 void atributo(char s1[],char s2[])
887 {
888     int i;
889     for(i=0;s2[i]!='\0';i++)
890         s1[i]=s2[i];
891     s1[i]='\0';
892 }
893
894 void atributo(char s1[],char s2[])
895 {
896     int i;
897     for(i=0;s2[i]!='\0';i++)
898         s1[i]=s2[i];
899     s1[i]='\0';
900 }
901
902 void atributo(char s1[],char s2[])
903 {
904     int i;
905     for(i=0;s2[i]!='\0';i++)
906         s1[i]=s2[i];
907     s1[i]='\0';
908 }
909
910 void atributo(char s1[],char s2[])
911 {
912     int i;
913     for(i=0;s2[i]!='\0';i++)
914         s1[i]=s2[i];
915     s1[i]='\0';
916 }
917
918 void atributo(char s1[],char s2[])
919 {
920     int i;
921     for(i=0;s2[i]!='\0';i++)
922         s1[i]=s2[i];
923     s1[i]='\0';
924 }
925
926 void atributo(char s1[],char s2[])
927 {
928     int i;
929     for(i=0;s2[i]!='\0';i++)
930         s1[i]=s2[i];
931     s1[i]='\0';
932 }
933
934 void atributo(char s1[],char s2[])
935 {
936     int i;
937     for(i=0;s2[i]!='\0';i++)
938         s1[i]=s2[i];
939     s1[i]='\0';
940 }
941
942 void atributo(char s1[],char s2[])
943 {
944     int i;
945     for(i=0;s2[i]!='\0';i++)
946         s1[i]=s2[i];
947     s1[i]='\0';
948 }
949
950 void atributo(char s1[],char s2[])
951 {
952     int i;
953     for(i=0;s2[i]!='\0';i++)
954         s1[i]=s2[i];
955     s1[i]='\0';
956 }
957
958 void atributo(char s1[],char s2[])
959 {
960     int i;
961     for(i=0;s2[i]!='\0';i++)
962         s1[i]=s2[i];
963     s1[i]='\0';
964 }
965
966 void atributo(char s1[],char s2[])
967 {
968     int i;
969     for(i=0;s2[i]!='\0';i++)
970         s1[i]=s2[i];
971     s1[i]='\0';
972 }
973
974 void atributo(char s1[],char s2[])
975 {
976     int i;
977     for(i=0;s2[i]!='\0';i++)
978         s1[i]=s2[i];
979     s1[i]='\0';
980 }
981
982 void atributo(char s1[],char s2[])
983 {
984     int i;
985     for(i=0;s2[i]!='\0';i++)
986         s1[i]=s2[i];
987     s1[i]='\0';
988 }
989
990 void atributo(char s1[],char s2[])
991 {
992     int i;
993     for(i=0;s2[i]!='\0';i++)
994         s1[i]=s2[i];
995     s1[i]='\0';
996 }
997
998 void atributo(char s1[],char s2[])
999 {
1000    int i;
1001    for(i=0;s2[i]!='\0';i++)
1002        s1[i]=s2[i];
1003    s1[i]='\0';
1004 }
1005
1006 void atributo(char s1[],char s2[])
1007 {
1008    int i;
1009    for(i=0;s2[i]!='\0';i++)
1010        s1[i]=s2[i];
1011    s1[i]='\0';
1012 }
1013
1014 void atributo(char s1[],char s2[])
1015 {
1016    int i;
1017    for(i=0;s2[i]!='\0';i++)
1018        s1[i]=s2[i];
1019    s1[i]='\0';
1020 }
1021
1022 void atributo(char s1[],char s2[])
1023 {
1024    int i;
1025    for(i=0;s2[i]!='\0';i++)
1026        s1[i]=s2[i];
1027    s1[i]='\0';
1028 }
1029
1030 void atributo(char s1[],char s2[])
1031 {
1032    int i;
1033    for(i=0;s2[i]!='\0';i++)
1034        s1[i]=s2[i];
1035    s1[i]='\0';
1036 }
1037
1038 void atributo(char s1[],char s2[])
1039 {
1040    int i;
1041    for(i=0;s2[i]!='\0';i++)
1042        s1[i]=s2[i];
1043    s1[i]='\0';
1044 }
1045
1046 void atributo(char s1[],char s2[])
1047 {
1048    int i;
1049    for(i=0;s2[i]!='\0';i++)
1050        s1[i]=s2[i];
1051    s1[i]='\0';
1052 }
1053
1054 void atributo(char s1[],char s2[])
1055 {
1056    int i;
1057    for(i=0;s2[i]!='\0';i++)
1058        s1[i]=s2[i];
1059    s1[i]='\0';
1060 }
1061
1062 void atributo(char s1[],char s2[])
1063 {
1064    int i;
1065    for(i=0;s2[i]!='\0';i++)
1066        s1[i]=s2[i];
1067    s1[i]='\0';
1068 }
1069
1070 void atributo(char s1[],char s2[])
1071 {
1072    int i;
1073    for(i=0;s2[i]!='\0';i++)
1074        s1[i]=s2[i];
1075    s1[i]='\0';
1076 }
1077
1078 void atributo(char s1[],char s2[])
1079 {
1080    int i;
1081    for(i=0;s2[i]!='\0';i++)
1082        s1[i]=s2[i];
1083    s1[i]='\0';
1084 }
1085
1086 void atributo(char s1[],char s2[])
1087 {
1088    int i;
1089    for(i=0;s2[i]!='\0';i++)
1090        s1[i]=s2[i];
1091    s1[i]='\0';
1092 }
1093
1094 void atributo(char s1[],char s2[])
1095 {
1096    int i;
1097    for(i=0;s2[i]!='\0';i++)
1098        s1[i]=s2[i];
1099    s1[i]='\0';
1100 }
1101
1102 void atributo(char s1[],char s2[])
1103 {
1104    int i;
1105    for(i=0;s2[i]!='\0';i++)
1106        s1[i]=s2[i];
1107    s1[i]='\0';
1108 }
1109
1110 void atributo(char s1[],char s2[])
1111 {
1112    int i;
1113    for(i=0;s2[i]!='\0';i++)
1114        s1[i]=s2[i];
1115    s1[i]='\0';
1116 }
1117
1118 void atributo(char s1[],char s2[])
1119 {
1120    int i;
1121    for(i=0;s2[i]!='\0';i++)
1122        s1[i]=s2[i];
1123    s1[i]='\0';
1124 }
1125
1126 void atributo(char s1[],char s2[])
1127 {
1128    int i;
1129    for(i=0;s2[i]!='\0';i++)
1130        s1[i]=s2[i];
1131    s1[i]='\0';
1132 }
1133
1134 void atributo(char s1[],char s2[])
1135 {
1136    int i;
1137    for(i=0;s2[i]!='\0';i++)
1138        s1[i]=s2[i];
1139    s1[i]='\0';
1140 }
1141
1142 void atributo(char s1[],char s2[])
1143 {
1144    int i;
1145    for(i=0;s2[i]!='\0';i++)
1146        s1[i]=s2[i];
1147    s1[i]='\0';
1148 }
1149
1150 void atributo(char s1[],char s2[])
1151 {
1152    int i;
1153    for(i=0;s2[i]!='\0';i++)
1154        s1[i]=s2[i];
1155    s1[i]='\0';
1156 }
1157
1158 void atributo(char s1[],char s2[])
1159 {
1160    int i;
1161    for(i=0;s2[i]!='\0';i++)
1162        s1[i]=s2[i];
1163    s1[i]='\0';
1164 }
1165
1166 void atributo(char s1[],char s2[])
1167 {
1168    int i;
1169    for(i=0;s2[i]!='\0';i++)
1170        s1[i]=s2[i];
1171    s1[i]='\0';
1172 }
1173
1174 void atributo(char s1[],char s2[])
1175 {
1176    int i;
1177    for(i=0;s2[i]!='\0';i++)
1178        s1[i]=s2[i];
1179    s1[i]='\0';
1180 }
1181
1182 void atributo(char s1[],char s2[])
1183 {
1184    int i;
1185    for(i=0;s2[i]!='\0';i++)
1186        s1[i]=s2[i];
1187    s1[i]='\0';
1188 }
1189
1190 void atributo(char s1[],char s2[])
1191 {
1192    int i;
1193    for(i=0;s2[i]!='\0';i++)
1194        s1[i]=s2[i];
1195    s1[i]='\0';
1196 }
1197
1198 void atributo(char s1[],char s2[])
1199 {
1200    int i;
1201    for(i=0;s2[i]!='\0';i++)
1202        s1[i]=s2[i];
1203    s1[i]='\0';
1204 }
1205
1206 void atributo(char s1[],char s2[])
1207 {
1208    int i;
1209    for(i=0;s2[i]!='\0';i++)
1210        s1[i]=s2[i];
1211    s1[i]='\0';
1212 }
1213
1214 void atributo(char s1[],char s2[])
1215 {
1216    int i;
1217    for(i=0;s2[i]!='\0';i++)
1218        s1[i]=s2[i];
1219    s1[i]='\0';
1220 }
1221
1222 void atributo(char s1[],char s2[])
1223 {
1224    int i;
1225    for(i=0;s2[i]!='\0';i++)
1226        s1[i]=s2[i];
1227    s1[i]='\0';
1228 }
1229
1230 void atributo(char s1[],char s2[])
1231 {
1232    int i;
1233    for(i=0;s2[i]!='\0';i++)
1234        s1[i]=s2[i];
1235    s1[i]='\0';
1236 }
1237
1238 void atributo(char s1[],char s2[])
1239 {
1240    int i;
1241    for(i=0;s2[i]!='\0';i++)
1242        s1[i]=s2[i];
1243    s1[i]='\0';
1244 }
1245
1246 void atributo(char s1[],char s2[])
1247 {
1248    int i;
12
```


Programa ejecutándose: capturas

```
C:\Users\nighth_000\Desktop\Diego de Compiladores\lab6.exe
Ingrese el identificador:123
Ingrese el atributo:dato2

Desea continuar(s/n)?s
Ingrese el identificador:1239
Ingrese el atributo:dato3

Desea continuar(s/n)?s
Ingrese el identificador:99999
Ingrese el atributo:dato4

Desea continuar(s/n)?n
La tabla de simbolo es:
Valores de la tabla      claves      atributos
0                        0
1                        12          dato1
2                        123         dato2
3                        0
4                        0
5                        0
6                        0
7                        1239        dato3
8                        0
9                        99999       dato4
10                       0

MENU de la tabla de simbolos:
1.crear una tabla de simbolos
2.Buscar en la tabla de simbolos
Ingrese su opcion: 2
Ingrese el atributo que quiera encontrar: dato4

El atributo o conjunto de simbolo --dato4-- esta presente en la tabla de simbolo con la clave:99999

MENU de la tabla de simbolos:
1.crear una tabla de simbolos
2.Buscar en la tabla de simbolos
Ingrese su opcion:
```