

## Esempio di cambio di tipo e utilizzo della tabella hash

Nel file **liste.h** si sostituisca la definizione di **Elem**:

```
typedef char * Elem;
```

con:

```
struct el_lista {
```

```
char* nome;
```

```
char* def;
```

```
};
```

```
typedef struct el_lista *
```

```
Elem;
```

Nel file **testTabHash.c** si sostituiscano le definizioni delle funzioni **confr**, **allCpy** e **libera** come segue

```
int confr(Elem x,Elem y) {  
  assert(x);  
  assert(y);  
  return strcmp(x->nome,y ->nome);}
```

```
Elem allCpy(Elem x) {  
  Elem nuovo;  
  nuovo = (Elem) malloc(sizeof(struct el_lista));  
  if (nuovo)  
  {nuovo ->nome = strAllCpy(x->nome);  
  nuovo ->def = strAllCpy(x->def);}  
  return nuovo;}
```

```
void libera(Elem x)  
{free(x->nome);  
free(x->def);  
free(x);}
```

## Esempio di cambio di tipo e utilizzo della tabella hash

```
int main()
{char temp1[MAX_LUN], temp2[MAX_LUN];
TabHash tabella;
struct el_lista app = {temp1,temp2};
Elem appP = &app;
int i,c;
tabella = costTabHash(DIM_TAB);
if (tabella)
{printf("inserisci %d coppie di stringhe di caratteri, di lunghezza minore di %d \n",NUM_EL,MAX_LUN);
for (i = 0;i < NUM_EL;i++)
{printf("inserisci le stringhe di caratteri\n");
printf("? \n");
scanf("%s%s", appP ->nome ,appP -> def);
c = insTab(tabella,appP);
if (c==1) printf("inserimento riuscito! \n"); else printf("inserimento non riuscito! \n");
}
i = cercaTab(tabella, appP);
if (i==1) {printf("i = %d, le stringhe %s e %s sono presenti \n",i,appP->nome,appP->def);
i = cancTab(tabella, appP);
if (i==1) printf("i = %d, le stringhe %s e %s sono presenti \n",i,appP->nome,appP->def);
else printf("rimozione fallita!");}
else printf("la stringa non è presente: errore!");
}
else printf("tabella vuota!");
return 0;}
```