**removefromhash**

*Removal of string identifier from hash table.*

**Syntax:**

**removefromhash**(*h\_id*, *str*)*;*

**Arguments:**

*h\_id –* identifier of hash table,

*str –* string identifier being deleted.

**Description:**

*removefromhash(h\_id, str)* – removal ofstring identifier *str* from hash table with identifier *h\_id*. This operation deletes all the records in hash table comprising string *str.*

Hash tables are intended for ensuring accelerated search for identifier in a list of strings.

**Result:**

*None*.

**Example:**

|  |  |
| --- | --- |
|  | **var**  s\_01:**string** = "string1" + chr(10) + "string2",  s\_02:**string** = "string1";  //creating list of strings  slist = **createstringlist**;  //load string s\_01 into list  **settextstringlist**(slist, s\_01);  //add string s\_02 into list  i = **addstringlist**(slist, s\_02, 0);  //create hash table  hasht = **createstringlisthash**(slist, 4095);  //delete string identifier  **removefromhash**(hasht, "string1");  //obtain string index  index = **getindexfromhash**(hasht, "string1");  //delete string  **freeobject(**slist**);**  //delete hash table  **freeobject(**hasht**);** |

In the course of performing example a list of strings with identifier *slist* will be createdwith the use of function *createstringlist*. A string declared with white-space character “string advance” will be loaded into a list of strings *slist* by means of function *settextstringlist*, as a result, variable *slist* will comprise an identifier of list of strings.

“” *string1*”

“*string2*””.

Further, string “*string1*” will be loaded into a list of strings with identifier *slist* by means of function *addstringlist*.

As a result, variable *slist* will comprise identifier of a list of strings.

“”string1”

“string2”

“string1””

, variable *i* will be assigned value 2.

Hash table with identifier *hasht* will be created by means of *createstringlisthash* function for a list of strings with identifier *slist*.

Use function *removefromhash* to delete string identifier “*string1*” from hash table with identifier *hasht.*

Index of string “string1” will be obtained in the list of strings *slist* by means of *getindexesfromhash* function, variable *index* will be assigned a value -1 (string is absent in hash table).

Prior to completing an example the list of strings and hash table shall be deleted by means of *freeobject* function.