```
 HASHY    EPL     03/08/66  1040.4

hash_table:  proc;
dcl (newkey, newvalue) fixed;
dcl n fixed;
    n = 8;  /*  this should be done with initial  */
dcl 1 table(n),
      2 (full, skip) bit(1),
      2 (key, value) fixed,
    ctl(p);
dcl q ptr;
dcl (error, delerr, failure) label;
dcl n_min fixed;
    n_min = 8;  /*  This should be done with "initial" attribute. */
dcl (lim1, lim2) float;
    lim1 = 0.33e0; lim2 = 0.67e0;  /*  initial... */
dcl (hash, i, j, m, old_n, new_n) fixed;

            /*  initialize table  */

call initialize;
return;

initialize: proc;
m = 0;
allocate p -¤g table set(p);
do i = 1 to n;
    p -¤g table(i).full, p -¤g table(i).skip = "0"b;
end;
return;
end initialize;


lookup:  entry(newkey, newvalue, error);
hash = mod(newkey, n-1);
loop1:
do i = hash to n, 1 to hash-1;
    if ¤n p -¤g table(i).full then
    do; if ¤n p -¤g table(i).skip then go to error; end;
    else if newkey = p -¤g table(i).key then
    do;
        newvalue = p -¤g table(i).value;
        return;
    end;
end loop1;
go to error;  /*  not found in full table - should be impossible  */


enter:  entry(newkey, newvalue, error);
hash = mod(newkey, n-1);
loop2:
do i = hash to n, 1 to hash-1;
    if ¤n p -¤g table(i).full then
    do;
        p -¤g table(i).key = newkey;
        p -¤g table(i).value = newvalue;
        p -¤g table(i).full = "1"b;
        m = m + 1;
        if m ¤g lim2*n then
        do;
```

```
            new_n = (lim2/lim1) * n;
            failure = error;
            go to rehash;
        end;
        else return;
    end;
end loop2;
go to error;  /*  table overflow - should be impossible  */


delete:  entry(newkey, delerr);
hash = mod(newkey, n-1);
loop3:
do i = hash to n, 1 to hash-1;
    if p ->g table(i).full then
    a3:
    do;
        if newkey = p ->g table(i).key then
        b3:
        do;
            p ->g table(i).full = "0"b;
            j = mod(i+1, n);
            p ->g table(i).skip =  /* mark skip */
                p ->g table(j).full ¢ p ->g table(j).skip;
            m = m - 1;
            if m ¬l lim1 * n then
            c3:
            do;
                new_n = (lim1/lim2) * n;
                if new_n ¬l n_min then return;
                failure = delerr;
                go to rehash;
            end c3;
            return;
        end b3;
    end a3;
    else
    if ¬n ->g table(i).skip then go to delerr;  /*  no entry found  */
end loop3;
go to delerr;  /* not found in full table - should be impossible */


rehash:
old_n = n;
n = new_n;
q = p;
allocate table set(q);
call initialize;
loop4:
do i = 1 to old_n;
    if q ->g table(i).full then
    call enter(q ->g table(i).key, q ->g table(i).value, failure);
end loop4;

free q ->g table;
return;

end hash_table;
```