Cs352 — Homework #4
SkipList

September 18, 2003

Due Time: 10/2/03 (9:00PM). Submission in pairs is allowed.

A SkipList works on a set $S = \{x_1 \dots x_n\}$ of *keys*, each of which (in this exercise) is an integer. It supports the following operations, which you have to implement:

Create(S) — Create an empty SkipList for S. Here we assume that S is empty (i.e., contains no elements)

Insert(x, S) Insert the key x into the data structure for S.

Delete(x, S) Delete the key x from S, if $x \in S$. If $x \notin S$, no action is needed.

Find(x, S) Report if x is in S.

Succ(x, S) Report the *successor* of x in S. The successor of x is defined as the smallest key in S which is strictly larger than x.

Your program, called *SLMaintaing* should read a file from the standard input. The file contains commands for the SkipList. Immediately after reading each command, you program should print the appropriate response. Every line in the file contains one command, and can be uniquely identified by the first character of the line. The format of the file is as follows:

% Every line that starts with the symbol % is a comment line. Its content (including the %) should be printed to the standard output, and no action should be taken on the SkipList itself.

C A line starting with 'C' should cause the creation of a new SkipList. Only one such line appears in the input file, and would appear at the beginning of the file.

I num Insert the key 'num' into the SkipList. After this command takes place, your program should print either "The key *num* was inserted into the SkipList" or "The key *num* already exists in the SkipList". The word "num" here stands for any integer within valid range.

D num Delete the key 'num' from the SkipList. After this command takes place, your program should print either "The key *num* was deleted from the SkipList" or "The key *num* does not exist in the SkipList".

F num Find the key *num*. After this command took place, your program should print either "The key *num* exists in the SkipList" or "The key *num* does not exist in the SkipList".

S num Find the successor of *num*. After this command takes place, your program should print either "The successor of *num* is *num2*" or "The key *num* does not have a successor in the SkipList".

P Prints all elements currently in the SkipList, level by level, starting with the top-most one. Do not print the keys with values INT_MIN and INT_MAX. The output of this command should be, for example

```
LEVEL 4 Keys: 23 149

LEVEL 3 Keys: 5 9 23 48 149 205

LEVEL 2 Keys: 2 5 9 23 27 48 149 159 164 192  205

LEVEL 1 Keys: 2 3 5 6 9 23 27  33 36 48 149 153 159 164 166 169 192 201 2021
```

An example of an input file is

```
%An example to the skiplist program
C
I 17
I 48
I 23
I 184
S 16
%Tra La la - I am a comment line
S 21
P
D 17
D 48
P
```

More comments

1. Since a SkipList is a randomized data structure, you are requested to use the function *OurRnd()* which will be available in the class directory cs352. This function returns the values 0 or 1 (False or True) each time it is called. Do not use any other randomized function. As described in class, during the insertion of a key x, we detemine the number k of levels that x participates in by the following simple loop.

```
k=1 ;
while(OurRnd()) k++ ;
```

2. Please use the following definitions

```
typedef struct cell { /*Basic cell of a SkipList */
    int key ;
    struct cell * nxt ;
    struct cell * down ;
} CELL ;
```

```
typedef struct sl { /*A header of a SkipList
                        Useful if more than one SkipList exists*/
    CELL * top ; /*points to the smallest element in the upper-most level */
    int n;  /* number of elements  */
    int l ; /* number of levels in the SL */
} SL ;
```