

Crayfish scrivener

ליאונרדו המציא מכונת כתיבה משוכללת בשם "הסרטן" (crayfish scrivener). למכונה יש שתי פקודות. פקודה אחת מוסיפה תו לסוף הטקסט. הפקודה השניה מבצעת UNDO ל - U הפעולות האחרונות. חשוב לשים לב שפקודת ה - UNDO היא בעצמה פקודה ולכן פקודת UNDO מאוחרת יותר יכולה לבטל אותה.

תיאור הבעיה

עליכם לכתוב תוכנית שמדמה את פעולת מכונת הסרטן: בהתחלה יש טקסט ריק. המערכת שולחת לתוכנית סדרה של פקודות ושאלות. בשאלתה, המערכת שואלת את התוכנית מה היא האות שנמצאת באינדקס מסוים של הטקסט, במצבו הנוכחי.

Init()

המערכת קוראת לפונקציה הזאת פעם אחת, בהתחלה. אתם יכולים להשתמש בפונקציה זו כדי לאתחל מבני נתונים.

TypeLetter(L)

המערכת קוראת לפונקציה הזאת כדי להפעיל את הפקודה שמוסיפה אות בסוף הטקסט הנוכחי. האות שתיכתב היא האות L. מובטח שזוהי אות קטנה באנגלית (בין a ל- z).

UndoCommands(U)

המערכת קוראת לפונקציה הזאת כדי להפעיל את פקודת ה- UNDO של מכונת הסרטן. לפקודה זו פרמטר U, והיא מחזירה את הטקסט למצב בו הוא היה לפני U פקודות.

GetLetter(P)

כשהמערכת קוראת לפונקציה הזאת, על הפונקציה לחשב ולהחזיר את האות שנמצאת באינדקס P בטקסט במצבו הנוכחי. האינדקס של האות הראשונה בטקסט הוא 0. השאלתה הזאת אינה פקודה למכונת הסרטן, והיא לא נספרת במניין הפקודות שפעולת UNDO מבטלת.

מובטח שבקריאה ל - UNDO, הפרמטר U לא יהיה גדול מהסך הכולל של פקודות שהתקבלו עד כה (להזכירכם, הפקודות האפשריות הן TypeLetter ו- UndoCommands). מובטח שבקריאה ל- TypeLetter, הפרמטר P יהיה קטן מאורך הטקסט הנוכחי.

לשם הבהרה, הסתכלו בדוגמא הבאה.

דוגמא

.We show a possible sequence of calls, together with the state of the text after each call

Current text	Returns	Call
		Init()
a		TypeLetter(a)
ab		TypeLetter(b)
ab	b	GetLetter(1)
abd		TypeLetter(d)
a		UndoCommands(2)
abd		UndoCommands(1)
abd	d	GetLetter(2)
abde		TypeLetter(e)
abd		UndoCommands(1)
ab		UndoCommands(5)
abc		TypeLetter(c)
abc	c	GetLetter(2)
abd		UndoCommands(2)
abd	d	GetLetter(2)

[Subtask 1 [5 points

The total number of commands and queries is between 1 and 100 (inclusive) and there will be no calls to `UndoCommands` ■

[Subtask 2 [7 points

The total number of commands and queries is between 1 and 100 (inclusive) and no `UndoCommands` will be undone ■

[Subtask 3 [22 points

.(The total number of commands and queries is between 1 and 5 000 (inclusive ■

[Subtask 4 [26 points

The total number of commands and queries is between 1 and 1 000 000 (inclusive). All calls to `GetLetter` will occur after all calls to `TypeLetter` and `UndoCommands` ■

[Subtask 5 [40 points

.(The total number of commands and queries is between 1 and 1 000 000 (inclusive ■

Implementation details

You have to submit exactly one file, called `scrivener.c`, `scrivener.cpp` or `scrivener.pas`. This file must implement the subprograms described above using the following signatures

C/C++ programs

```
void Init();
void TypeLetter(char L);
void UndoCommands(int U);
char GetLetter(int P);
```

Pascal programs

```
procedure Init;
procedure TypeLetter(L : Char);
procedure UndoCommands(U : LongInt);
function GetLetter(P : LongInt) : Char;
```

These subprograms must behave as described above. Of course you are free to implement other subprograms for their internal use. Your submissions must not interact in any way with standard .input/output, nor with any other file

Sample grader

:The sample grader reads the input in the following format

- line 1: the total number of commands and queries in the input ■
- on each following line ■
- ;T followed by a space and a lowercase letter for a `TypeLetter` command ■
- ;U followed by a space and an integer for `UndoCommands` ■
- .P followed by a space and an integer for `GetLetter` ■

.The sample grader will print the characters returned by `GetLetter`, each one on a separate line