International Olympiad in Informatics 2012
23-30 September 2012
Sirmione - Montichiari, Italy
Competition tasks, day 1: Leonardo's inventions and projects

scrivener
ไทย — 1.1

กุ้งนักจด

มีคนกล่าวไว้ว่าลีโอนาร์โดนั้นชื่นชมโยฮัน กูเต็นเบิร์ก (ช่างตีเหล็กชาวเยอรมันผู้ซึ่งเป็นคนค้นคิดการจัดเรียงพิมพ์) เป็นอย่างมาก และเขาได้สร้างเครื่องพิมพ์อย่างง่ายชื่อ กุ้งนักจด (il gambero scrivano) ขึ้นมาเพื่อเป็นการระลึกถึงกูเต็นเบิร์ก กุ้งนักจดนั้นมีความคล้ายคลึงกับเครื่องพิมพ์ดีดในสมัยปัจจุบัน มันรับคำสั่งได้เพียงสองคำสั่ง คือคำสั่งพิมพ์ตัวอักษร และคำสั่งยกเลิกคำสั่งล่าสุด จุดเด่นของ กุ้งนักจด คือคำสั่งยกเลิกนั้นเจ๋งมาก กล่าวคือ คำสั่งยกเลิกนั้นถือเป็นคำสั่งเช่นกันและสามารถถูกยกเลิกได้เช่นกัน

ปัญหา

หน้าที่ของคุณคือสร้าง กุ้งนักจด เวอร์ชั่นซอฟต์แวร์ขึ้นมา มันจะเริ่มต้นด้วยข้อความว่าง และรับลำดับคำสั่งจากผู้ใช้ และรับคำถามสำหรับตำแหน่งที่เจาะจงมาของข้อความปัจจุบัน ดังต่อไปนี้

- `Init()` — ถูกเรียกเพียงครั้งเดียวเมื่อเริ่มทำงานโดยไม่มี argument ใด ๆ คำสั่งนี้สามารถใช้ในการ initialize ค่าเริ่มต้นของโครงสร้างข้อมูล มันจะไม่มีการขอยกเลิกคำสั่งนี้
- `TypeLetter(L)` — เพิ่มอักษรตัวเล็ก L ไปด้านท้ายของข้อความ อักษร L นั้นมาจาก `a`, ..., `z`.
- `UndoCommands(U)` — ยกเลิก U คำสั่งล่าสุด โดย U มีค่าเป็นจำนวนเต็มบวก
- `GetLetter(P)` — คืนค่าตัวอักษร ณ ตำแหน่ง P ในข้อความปัจจุบัน โดย P เป็นตำแหน่งมีค่าไม่เป็นลบ ตัวอักษรแรกในข้อความมีตำแหน่งเป็น 0 (คำถามนี้ไม่นับเป็นคำสั่ง คำสั่งยกเลิกจะไม่สนใจคำถามนี้)

หลังจากการเรียก `Init` ครั้งแรก ฟังก์ชันย่อยอื่น ๆ จะถูกเรียกใช้ 0 ครั้งหรือมากกว่าในลำดับใดก็ได้ รับประกันว่า U จะไม่เกินจำนวนของคำสั่งที่ได้รับมาแล้ว และ P จะมีค่าน้อยกว่าความยาวของข้อความปัจจุบัน (น้อยกว่าจำนวนของตัวอักษรในข้อความปัจจุบัน)

คำสั่ง `UndoCommands(U)` จะยกเลิกคำสั่ง U คำสั่งก่อนหน้าตามลำดับ ย้อนหลัง ถ้าคำสั่งที่จะถูกยกเลิกคือ `TypeLetter(L)` มันจะเอาตัวอักษร L ออกจากด้านท้ายของข้อความ ถ้าคำสั่งที่จะถูกยกเลิกคือ `UndoCommands(X)` สำหรับค่า X ใด ๆ มันจะทำคำสั่ง X คำสั่งก่อนหน้าตามลำดับ ปรกติ ที่ได้รับ

ตัวอย่าง

ขอแสดงลำดับการเรียกใช้ฟังก์ชันย่อยต่าง ๆ รวมถึงสถานะของข้อความหลังจากการเรียกใช้ฟังก์ชันย่อยแต่ละอัน

คำสั่ง	ค่าที่คืนกลับมา	ข้อความปัจจุบัน
Init()		
TypeLetter(a)		a
TypeLetter(b)		ab
GetLetter(1)	b	ab
TypeLetter(d)		abd
UndoCommands(2)		a
UndoCommands(1)		abd
GetLetter(2)	d	abd
TypeLetter(e)		abde
UndoCommands(1)		abd
UndoCommands(5)		ab
TypeLetter(c)		abc
GetLetter(2)	c	abc
UndoCommands(2)		abd
GetLetter(2)	d	abd

งานย่อยที่ 1 [5 แต้ม]

- จำนวนรวมของคำสั่งและคำถามอยู่ระหว่าง 1 ถึง 100 (รวมหัวท้าย) และไม่มีการเรียกใช้คำสั่ง `UndoCommands`

งานย่อยที่ 2 [7 แต้ม]

- จำนวนรวมของคำสั่งและคำถามอยู่ระหว่าง 1 ถึง 100 (รวมหัวท้าย) และจะไม่มีการยกเลิกคำสั่ง `UndoCommands`

งานย่อยที่ 3 [22 แต้ม]

- จำนวนรวมของคำสั่งและคำถามอยู่ระหว่าง 1 ถึง 5,000 (รวมหัวท้าย)

งานย่อยที่ 4 [26 แต้ม]

- จำนวนรวมของคำสั่งและคำถามอยู่ระหว่าง 1 ถึง 1,000,000 (รวมหัวท้าย) คำสั่ง `GetLetter` ทั้งหมด จะเกิดขึ้นหลังจากคำสั่ง `TypeLetter` และ `UndoCommands` ทั้งหมด

งานย่อยที่ 5 [40 แต้ม]

- จำนวนรวมของคำสั่งและคำถามอยู่ระหว่าง 1 ถึง 1,000,000 (รวมหัวท้าย)

รายละเอียดสำหรับการเขียนโปรแกรม

คุณจะต้องส่งไฟล์หนึ่งไฟล์ในชื่อ `scrivener.c`, `scrivener.cpp` หรือ `scrivener.pas` ไฟล์นี้จะต้องเขียนโปรแกรมย่อยดังที่กล่าวไว้ข้างต้นโดยใช้รูปแบบดังต่อไปนี้

ภาษา C/C++

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

ภาษา Pascal

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

โปรแกรมย่อยเหล่านี้จะต้องทำงานตามที่ได้ระบุไว้ข้างต้น คุณสามารถเขียนโปรแกรมย่อยอื่น ๆ สำหรับใช้งานได้ โปรแกรมของคุณจะต้องไม่ยุ่งเกี่ยวกับ standard input/output หรือกับไฟล์ใด ๆ

grader ตัวอย่าง

grader ตัวอย่าง จะอ่าน Input ดังรูปแบบต่อไปนี้

- บรรทัดที่ 1: จำนวนรวมของคำสั่งและคำถามใน Input
- ในแต่ละบรรทัดหลังจากนั้น
  - `T` ตามด้วยข่องว่างและอักษรตัวเล็กสำหรับคำสั่ง `TypeLetter`
  - `U` ตามด้วยข่องว่างและตัวเลขจำนวนเต็มสำหรับคำสั่ง `UndoCommands`
  - `P` ตามด้วยข่องว่างและตัวเลขจำนวนเต็มสำหรับคำสั่ง `GetLetter`

grader ตัวอย่างจะพิมพ์ตัวอักษรที่คืนค่ามาจาก `GetLetter` บรรทัดละหนึ่งตัวอักษร