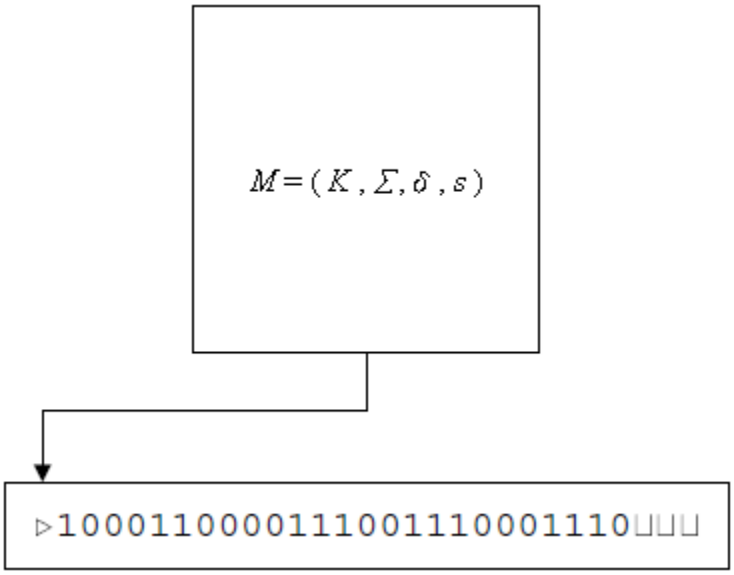
題目 H
圖靈機

執行時間限制: 10 **秒**

在資訊界中，P 是否就是 NP 是一個有名的未解問題。然而我們可以不正式地定義 P 及 NP ，如果一個問題能讓電腦在多項式時間內解決，那麼這類題目就屬於 P；如果一個題目找到了答案，且可以在多項式時間內驗證這個答案是否正確，則這類的題目就是 NP。然而精準的定義，卻有一套龐大的理論支持著，而這套理論中，最基本的模型就是圖靈機(Turing Machine)。

什麼是圖靈機呢？它的樣子就長得如下，上面的方框框就好像電腦的CPU，他負責運算下一個指令會是什麼。下面有一條帶子存放著字串，這個帶子的長度是無限的，而帶子就好像是記憶體，記錄著現在的運算結果。中間有一個箭頭指到下面的帶子，讓上面的框框能知道目前所指的地方是什麼字母。



讓我們定義圖靈機：一個圖靈機 M 包含著四樣東西，即 $M = (K, \Sigma, \delta, s)$。讓我們一項一項解釋：

- K 爲一個狀態的集合，它定義這個圖靈機有哪些狀態，且一個合理的圖靈機一定有 yes 和 no 這兩種狀態，當圖靈機的狀態切換到 yes，代表所給定的輸入字串是合理的，反之當狀態切到 no，則代表所給定的輸入字串是錯誤的。

- Σ 是一個字母的集合，代表帶子裡可以寫進哪些字母，以上面的例子來講，Σ 有四種字母 $(\triangleright, \sqcup, 0, 1)$，其中 $\triangleright$ 爲開頭字母，$\sqcup$ 爲空字母，這兩個特殊字母也是每個圖靈機必備的。

- δ 是一個函數，它是整個圖靈機的靈魂，負責決定當在什麼狀況會發生什麼事。δ 函數長的像這個樣子：

  δ(現在的狀態, 箭頭指到的字母)
  = (下一個狀態, 在箭頭上指到的地方要被替換成這個字母, 箭頭怎麼走)

  若用一般的話來解釋，就是當圖靈機在某個狀態且箭頭指到某個字母時，它會做三件事情，換成下一個狀態，將箭頭上的字母換掉，將箭頭移動。而箭頭只有三種移法：向左一格，向右一格及維持原地，我們以 $\leftarrow, \rightarrow, -$ 來表示。

- s 爲一開始的狀態，且不會是 yes 或 no。

爲了讓大家了解圖靈機的實際運作，我們以一個能判斷輸入字串長度是否爲偶數的圖靈機當作例子：

- $K = \{yes, no, start, odd, even\}$，也就是說這個圖靈機總共有五種狀態。

- $\Sigma = \{\triangleright, \sqcup, 1\}$，也就是除了圖靈機內設的兩個字母外，這個圖靈機只有額外的一個字母1。

- δ 函數將有 3×3 條 (三種可能的狀態，箭頭可能指到三種字母)：
  - $\delta(start, \triangleright) = (even, \triangleright, \rightarrow)$
  - $\delta(start, \sqcup) = (no, \sqcup, \rightarrow)$
  - $\delta(start, 1) = (no, \sqcup, \rightarrow)$

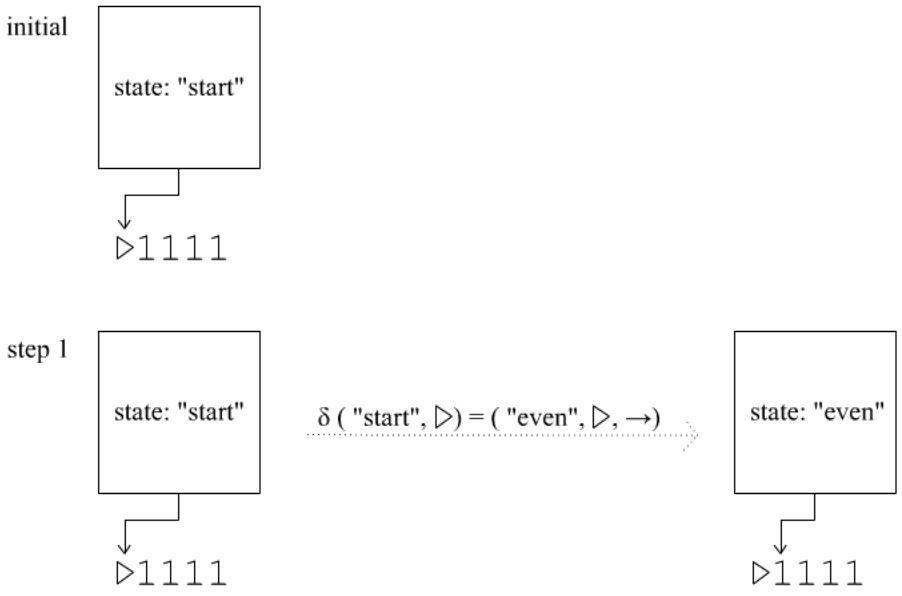
- $\delta(odd, \triangleright) = (no, \sqcup, \rightarrow)$
- $\delta(odd, \sqcup) = (no, \sqcup, \rightarrow)$
- $\delta(odd, 1) = (even, \triangleright, \rightarrow)$
- $\delta(even, \triangleright) = (no, \sqcup, \rightarrow)$
- $\delta(even, \sqcup) = (yes, \sqcup, \rightarrow)$
- $\delta(even, 1) = (odd, \sqcup, \rightarrow)$

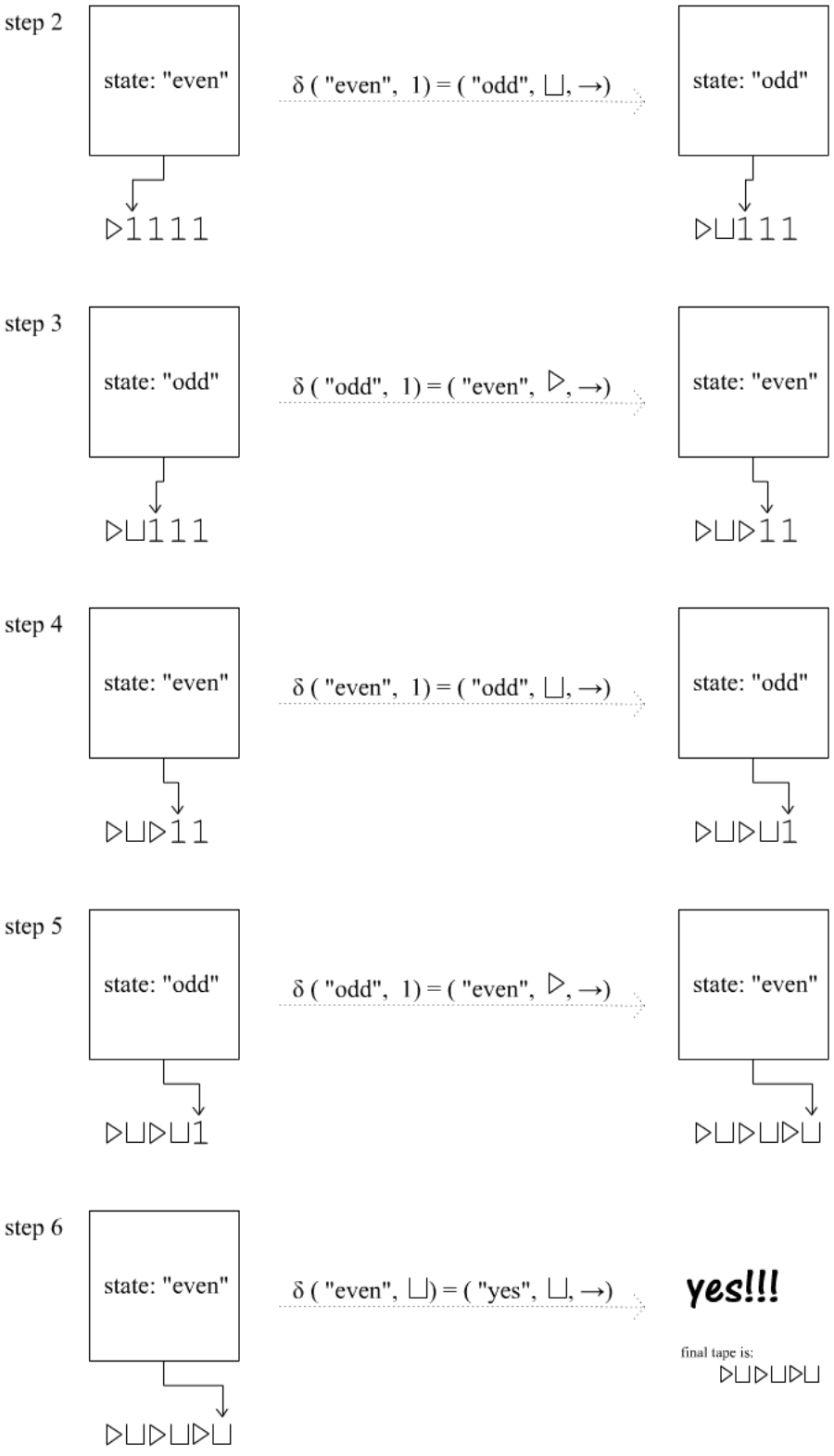
  其中你將會發現第二、三、四、七條是完全不會用到的。而當經過 δ 函數得到狀態爲 yes 或 no 時，其實圖靈機就結束了，所以箭頭取代的字及箭頭移動方向也就不重要。

- s 設爲 $start$。

定義好了圖靈機，接著我們就要將輸入字串丟進這個圖靈機。對於圖靈機的輸入字串，一定要用 Σ 裡面的字母，但不可以使用 $\triangleright, \sqcup$ 這兩項。我們假設輸入字串爲 1111，那麼我們要做的第一件事情，就是將輸入字串丟到圖靈機的帶子裡，做法是這樣：先將第一個字母設成 $\triangleright$，接著放入 1111，因此帶子長度爲 5。

我們將圖靈機的輸入字串放入帶子後，就可以開始執行。




step 2
state: "even"
δ ("even", 1) = ("odd", ␣, →)
state: "odd"
▷1111
▷␣111
step 3
state: "odd"
δ ("odd", 1) = ("even", ▷, →)
state: "even"
▷␣111
▷␣▷11
step 4
state: "even"
δ ("even", 1) = ("odd", ␣, →)
state: "odd"
▷␣▷11
▷␣▷␣1
step 5
state: "odd"
δ ("odd", 1) = ("even", ▷, →)
state: "even"
▷␣▷␣1
▷␣▷␣▷␣
step 6
state: "even"
δ ("even", ␣) = ("yes", ␣, →)
yes!!!
final tape is:
▷␣▷␣▷␣
▷␣▷␣▷␣

你會發現一件事情，在 step 5 的時候，δ 函數指示要往右走，但原本帶子右邊是沒東西的，當這種情況發生時，圖靈機會自動的生出一個 $\sqcup$ 讓長度變成 6，但反之若箭頭往左移，圖靈機的帶子並不會變短。

當圖靈機發現下一個狀態是 yes 或者 no 時，便不會再繼續運作。以上圖的 step 6 作例子，當看到狀態是 yes 時，箭頭上所指的字母不會被替換，且箭頭不會往右移 (這也是爲什麼長度還是 6 的原因)。

而這個圖靈機中，若你給奇數個 1，最後的狀態會是 no。

你的任務就是要模擬不同的圖靈機。

■ 輸入檔說明

第一行有一個整數 T ($T \leq 20$)，代表接下來有幾組測試資料。

每組測試資料代表一個圖靈機，對於每個圖靈機，會有很多行來表示它。第一行有兩個正整數 S, M，其中 S 代表圖靈機裡的 K 有幾個狀態，爲了簡化我們設狀態的名稱爲 1 到 S，且 1 代表 yes，2 代表 no，3 代表 $start$ ，也就是圖靈機裡的 s，因此題目一定保證 $S \geq 3$，同時也保證 $S \leq 100$。而 M 代表圖靈機裡有 $M + 2$ 個字母，分別爲 $\triangleright, \sqcup, a, b, \cdots$，例如 $M = 3$，則有 5 個字母 $\triangleright, \sqcup, a, b, c$。爲了輸入，我們以 s 代替 $\triangleright$，u 代替 $\sqcup$，而 M 最大值爲 13，也就是其它字母只會用到 a 至 m。

接下來的 $(S - 2) \times (M + 2)$ 行，定義著 δ 函式，每一行有五樣東西 P, A, Q, B, D，其中 P 爲一正整數，代表現在的狀態。A 爲一字元，代表箭頭指到的字母。Q 爲一正整數，代表要轉換到的下一個狀態。B 爲一字元，代表箭頭所指到的字母要替換成 B。最後 D 爲一字元，可能爲 $<, >, -$ (小於，大於，減號)，代表箭頭向左移動一格，向右移動一格或不動。

接著爲一正整數 C，代表對這個圖靈機有幾份需要模擬的輸入。最後有 C 行字串，代表要詢問圖靈機的輸入，每行字串最長爲 1000 個字元。請注意，輸入有可能是空字串。測資保證每一組圖靈機間不會有其它空白行。

測資不會讓圖靈機產生無窮迴圈的狀況，每個輸入字串最多只會讓所屬的圖靈機在 100000 步內的模擬保證有結果。測資不會有箭頭在最左邊時，卻還下往左的指令。

■ 輸出檔說明

對於每一個輸入字串，如果結果是 yes，請輸出 `yes xxxxx`，其中 `xxxxx` 是最後帶子上的字串。如果結果是 no，請輸出 `no`。

■ 範例輸入

```
2
5 1
3 s 5 s >
3 u 2 u >
3 a 2 u >
4 s 2 u >
4 u 2 u >
4 a 5 s >
5 s 2 u >
5 u 1 u >
5 a 4 u >
2
aaaa
aaaaaa
5 1
3 s 5 s >
3 u 2 u >
3 a 2 u >
4 s 2 u >
4 u 2 u >
4 a 5 s >
5 s 2 u >
5 u 1 u >
5 a 4 u >
1
aaa
```

■ 範例輸出

```
yes sususu
yes susususu
no
```