Algoritmica - Prova di Laboratorio del ?/0?/20??

Risolvete il seguente esercizio, prestando particolare attenzione alla formattazione dell'input e dell'output, in quanto la correzione è automatica.
Per consegnare un elaborato dovete fornire il codice sorgente attraverso il comando ./consegna che avete nella vostra home directory. Il comando deve essere utilizzato nel seguente modo:

./consegna *fileSorgente.c numEx*

ricordando che il percorso deve essere specificato a partire dalla vostra home directory e numEx deve essere un identificativo numerico (es. 1).
Il comando **consegna** *può essere utilizzato molteplici volte, per cui è possibile sovrascrivere la propria soluzione per un dato esercizio. Di tutte le consegne per un dato esercizio, viene corretta soltanto l'ultima.*
Il file da consegnare deve contenere nelle prime righe un commento C che specifica il vostro Nome, Cognome e Numero di Matricola. Per esempio:

```
/*
  Nome: Alan
  Cognome: Turing
  Matricola: 193700
*/
```

File non contenenti tali informazioni NON saranno ritenuti validi.

Nota: *E' possibile consultare i manuali (in inglese) contenenti la spiegazione di funzionamento e la sintassi di funzioni di libreria utilizzando il comando* **man**, *ad esempio:*

man strlen

Esercizio 1

Scopo dell'esercizio è trovare occorrenze di stringhe particolari detti *2-pattern* all'interno di un testo dato. Un 2-pattern è definito come una stringa formata soltanto da lettere minuscole e da una occorrenza al suo interno del carattere *. I 2-pattern rappresentano in modo compatto un insieme di stringhe: ogni occorrenza di '*' rappresenta una qualsiasi sottostringa (anche vuota) composta da lettere minuscole. Un 2-pattern P descrive quindi un insieme infinito $S(P)$ di stringhe. Ad esempio, il pattern $P =$"aba*cd" rappresenta tutte le stringhe che iniziano con "aba" e finiscono con "cd", tra cui possiamo contare "abacd" come "abaaaabacd". Al contrario, "abcd" non fa parte di $S(P)$.

Sia T una stringa normale che rappresenta un testo, e sia P un 2-pattern. Se T contiene una sottostringa t che inizia alla posizione k e che appartiene all'insieme di stringhe $S(P)$, allora si dice che T *soddisfa* P in posizione k. L'esercizio prevede, dati P e T, di elencare tutte le posizioni in cui T soddisfa S, qualora esse esistano. Ad esempio "braccobaldo" soddisfa il pattern b*o" in posizione 0 e in posizione 6, corrispondenti alle 'b'.

Scrivete un programma che prenda in input un 2-pattern P, un testo T e elenchi tutte le posizioni in cui T soddisfa P secondo le regole espresse nella sezione che indica il formato di output.

L'input e' formattato nel seguente modo. Il pattern $P =$ "$\mathcal{A}$*$\mathcal{B}$" dove $\mathcal{A}$ e $\mathcal{B}$ sono le due parti non *, è descritta nel file di input attraverso le sole parti $\mathcal{A}$ e $\mathcal{B}$. Il file di input quindi contiene $\mathcal{A}$, la prima parte di P sulla prima riga. La seconda riga contiene $\mathcal{B}$. La terza riga contiene il testo T.

Si può assumere che sia le due parti di P e T non contengano altro che lettere minuscole (non ci sono spazi o a-capo). Inoltre, si assuma che $\mathcal{A}$ e $\mathcal{B}$ di P in totale non superino 32 caratteri e che T non superi i 4096 caratteri. Si garantisce che $\mathcal{A}$ e $\mathcal{B}$ non siano mai vuote.

L'output **deve** contenere **solo** e **soltanto** una sequenza di interi, separati da uno spazio. L'i-esimo intero della sequenza contiene la posizione dell'i-esima volta in cui T soddisfa P. L'output deve rispettare i seguenti vincoli:

- Le posizioni in cui T soddisfa P devono essere elencate da sinistra verso destra, in ordine crescente di posizione.
- La posizione 0 corrisponde al primo carattere del testo.
- La stessa posizione non deve mai comparire più di una volta in output.
- L'output deve terminare con un a-capo ('\ n') al termine della sequenza.

Esempio

Input

```
al
er
fdalbertoxambadapalerermito
```

Output

```
2 17
```