

Università di Catania
Dipartimento di Matematica e Informatica
Corso di Studio in Informatica, A.A. 2025-2026
Prova di Laboratorio
Gennaio 2026

Descrizione del programma

Scrivere un programma in C che:

- A. Prenda un input da tastiera (argomenti della funzione main) costituito dal nome di un file di input (es: input.txt), il nome di un file di output (es: output.txt), e due parametri positivi n ed m , con $n < m$;
- B. Legga il file di input, il quale contiene un certo numero di frasi in italiano, separate da ritorni a capo. La lunghezza di ogni frase è minore o uguale a 200 caratteri.
- C. Inserisca in uno stack (implementato come struttura dati dinamica) i record corrispondenti alle sole frasi che contengono un numero di parole W , tale che $n \leq W \leq m$. Ogni record dovrà contenere due campi: la frase (escluso il carattere di ritorno a capo) e la lunghezza della frase stessa.
- D. Rimuova dallo stack tutti i record, uno per volta, e stampi sullo standard output i record stessi nel seguente modo (dove L è la lunghezza della frase e W il numero di parole):

W ## L ## FRASE
- E. Contestualmente alla rimozione del singolo record dallo stack, la frase contenuta nel record sia riversata nel file di output, una frase per riga, ed inoltre ogni lettera vocale della frase sia sostituita con la lettera 'X'.

Specifiche

Il programma potrà essere strutturato in un unico file sorgente, ma dovrà contenere almeno le seguenti funzioni:

1. **[3 pt] readInput():** funzione che prende in input l'array di puntatori a carattere argv ed il numero di argomenti argc della funzione main, controlla che gli argomenti richiesti siano nel numero e nei limiti specificati, e restituisca i parametri {finput,foutput,n,m} in una struct da restituire al chiamante.
2. **[4 pt] countWords():** funzione che prende in input una stringa (frase) e restituisce il numero di parole contenute nella frase (si assuma che le parole siano separate da singoli spazi);
3. **[3 pt] push():** funzione che prende in input una frase ed inserisce un opportuno record (lunghezza, frase) nello stack;
4. **[3 pt] pop():** funzione che preleva il record al top dello stack e restituisce la frase contenuta nel record;
5. **[6 pt] buildStack():** funzione con opportuni parametri formali, che legga tutte le frasi dal file di input e le inserisca in un opportuno stack di record, come descritto nei punti B e C;
6. **[3 pt] elabStrVowels():** funzione che sostituisce tutte le vocali presenti in una stringa (frase) con un carattere specificato come parametro;
7. **[2 pt] printRecord():** funzione che stampa un singolo record sullo standard output come specificato al punto D;
8. **[6 pt] clearStack():** funzione che rimuove tutti i record dallo stack, stampa i record come specificato al punto D, e salva la frase del record nel file di testo di output come specificato nel punto E.

È VIETATO usare variabili globali.

Durata della prova: 120 minuti. NB: Inserire nome, cognome e numero di matricola all'interno del file sorgente.

Output di controllo:

```
./a.out input.txt out.txt 10 15
```

```
65 ## 10 ## Capi che non serviva una risposta, bastava smettere di scappare.
71 ## 13 ## Fece un passo avanti senza sapere bene perché, ed era questo il
bello.
72 ## 14 ## La notte sembrava non finire, ma forse era solo lui a non voler
dormire.
68 ## 12 ## Aprì il quaderno e trovò parole che non ricordava di aver scritto.
67 ## 10 ## Ogni volta che tornava lì, qualcosa era cambiato, soprattutto lui.
79 ## 13 ## Si sedette sul marciapiede, convinto che da lì il mondo sembrasse
più onesto.
91 ## 13 ## La musica usciva gracchiante dalle cuffie, perfetta colonna sonora
per quel momento storto.
76 ## 14 ## Disse di stare bene, ma il modo in cui evitava gli sguardi raccontava
altro.
55 ## 10 ## Il vento gli scompigliava i pensieri più dei capelli.
72 ## 14 ## Non era il giorno giusto per fare l'eroe, e lo sapeva fin troppo
bene.
69 ## 12 ## Chiuse la porta piano, come se il rumore potesse svegliare i ricordi.
78 ## 12 ## Il messaggio arrivò in ritardo, ma bastò a cambiare completamente
la serata.
71 ## 12 ## Rise troppo forte, forse per coprire il silenzio che gli pesava
dentro.
91 ## 15 ## La casa scricchiolava come se volesse parlare, ma lui preferì
ignorarla e salire le scale.
91 ## 14 ## Aveva promesso di non pensarci più, eppure quel nome tornava sempre
nei momenti sbagliati.
91 ## 14 ## Le scarpe erano fradice, l'umore pure, ma continuò a camminare senza
voltarsi indietro.
```

Contenuto del file out.txt

```
CXpì chX nXn sXrvXvX XnX rXspXstX, bXstXvX smXttXrX dX scXppXrX.
FXcX Xn pXssX XvXntX sXnzX sXpXrX bXnX pXrché, Xd XrX qXXstX Xl bXllX.
LX nXttX sXmbrXvX nXn fXnXrX, mX fXrsX XrX sXlX lXX X nXn vXlXr dXrmXrX.
```

Aprì Xl qXXdXrnX X trXvò pXrXlX chX nXn rXcXrdXvX dX XvXr scrXttX.
OgnX vXltX chX tXrnXvX lì, qXXlcXsX XrX cXmbXXtX, sXprXttXttX lXX.
SX sXdXttX sXl mXrcXXpXXdX, cXnvXntX chX dX lì Xl mXndX sXmbrXssX pXù
XnXstX.
LX mXsXcX XscXvX grXcchXXntX dXllX cXffXX, pXrfXttX cXlXnnX sXnXrX pXr
qXXl mXmXntX stXrtX.
DXssX dX stXrX bXnX, mX Xl mXdX Xn cXX XvXtXvX glX sgXXrdX rXccXntXvX
XltrX.
Il vXntX glX scXmpXglXXvX X pXnsXXrX pXù dXX cXpXllX.
NXn XrX Xl gXXrnX gXXstX pXr fXrX l'XrXX, X lX sXpXvX fXn trXppX bXnX.
ChXXsX lX pXrtX pXXnX, cXmX sX Xl rXmXrX pXtXssX svXglXXrX X rXcXrdX.
Il mXssXggXX XrrXvò Xn rXtXrdX, mX bXstò X cXmbXXrX cXmplXtXmXntX lX
sXrXtX.
RXsX trXppX fXrtX, fXrsX pXr cXprXrX Xl sXlXnzXX chX glX pXsXvX dXntrX.
LX cXsX scrXcchXXlXvX cXmX sX vXlXssX pXrlXrX, mX lXX prXfXrì XgnXrXrlX
X sXlXrX lX scXlX.
AvXvX prXmXssX dX nXn pXnsXrcX pXù, XppXrX qXXl nXmX tXrnXvX sXmprX nXX
mXmXntX sbXglXXtX.
LX scXrpX XrXnX frXdXcX, l'XmXrX pXrX, mX cXntXnXò X cXmmXnXrX sXnzX
vXltXrsX XndXXtrX.