

Esame 20240725

Esercizio 1

(1) Esercizio 1 v1

ESSAY marked out of 10 penalty 0 File picker

Si scriva un programma C++ che legga un file di testo il cui nome è' ottenuto da riga di comando. Il programma opera come segue:

- Definisce un array `text` di 5000 `char`.
- Definisce un array `pwords` di 5000 puntatori a `char` e inizializza a `NULL` ciascun suo elemento.
- Apre in lettura il file specificato a linea di comando (ad esempio il file "testo.txt"), nel caso il file non esista il programma deve ritornare un messaggio di errore e terminare.
- Finché il file non è finito o si sono raggiunti i 5000 caratteri letti, legge ciascun carattere presente nel file utilizzando la funzione di libreria `get`. Ad esempio, se si è chiamato `fin` lo stream associato al file e `c` è una variabile di tipo `char`, la chiamata alla funzione `get` avviene in questo modo: `fin.get(c)`.
- Copia il carattere appena letto nell'array `text` e, se il carattere corrisponde alla prima lettera di una parola, assegna all'elemento corrente dell'array `pwords` l'indirizzo dell'elemento dell'array `text` in cui è appena stato copiato il carattere. Per semplicità, si assuma che un carattere rappresenta la prima lettera di una parola se esso è una lettera e se il carattere precedente è uno spazio o il carattere new line (`'\n'`).
- Chiama una funzione `ContaParole` che riceve come parametro un array di puntatori a `char` e restituisce un intero che corrisponde al numero di parole trovate.
- Chiama inoltre una funzione `GeneraParole` che riceva come parametro un array di puntatori a `char`, un numero intero che è il valore di ritorno di `ContaParole`, e un array di 7 `char`. La funzione costruisce una parola di 6 `char` ottenuti estraendo in modo casuale 6 delle lettere iniziali puntate dai puntatori non nulli contenuti nell'array passato alla funzione. La funzione non ritorna alcun valore, ma popola correttamente l'array di 7 `char` (inserendo il terminatore di stringa nell'ultimo elemento).
- Al termine della lettura del file chiude il file e stampa a video (1) il testo letto in input (cioè il contenuto dell'array `text`), (2) il numero di parole trovate (cioè il numero di elementi di `pwords` diverso da `NULL`), (3) tutte le prime lettere delle parole trovate nel testo (cioè il carattere puntato da ogni elemento di `pwords` diverso da `NULL`), e (4) la stringa popolata da `GeneraParole`.

Il file `esercizio1.cpp` contiene gli include che definiscono le funzioni consentite ed è da completare con quanto richiesto sopra. Di seguito è riportato un esempio di esecuzione (la formattazione dell'output deve essere rispettata).

```
marco > ./a.out testo.txt
Text: Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura, ch  la diritta via era smarrita

Number of words: 19
Initials: Nmddcnvmrpusocldves
Generated word: mlddpm
marco > ./a.out testononesiste.txt
Cannot open the input file "testononesiste.txt"
```

Note:

- Scaricare il file `esercizio1.cpp`, modificarlo per inserire il codice necessario per rispondere a questo esercizio. **Caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`, `fstream`, `cstdlib`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).

`esercizio1.cpp`

`testo.txt`

Information for graders:

(2) Esercizio 1 v2

ESSAY marked out of 10 penalty 0 File picker

Si scriva un programma C++ che legga un file di testo il cui nome è' ottenuto da riga di comando. Il programma opera come segue:

- Definisce un array `text` di 5000 `char`.
- Definisce un array `pwords` di 5000 puntatori a `char` e inizializza a `NULL` ciascun suo elemento.
- Apre in lettura il file specificato a linea di comando (ad esempio il file "testo.txt"), nel caso il file non esista il programma deve ritornare un messaggio di errore e terminare.
- Finché il file non è finito o si sono raggiunti i 5000 caratteri letti, legge ciascun carattere presente nel file utilizzando la funzione di libreria `get`. Ad esempio, se si è chiamato `fin` lo stream associato al file e `c` è una variabile di tipo `char`, la chiamata alla funzione `get` avviene in questo modo: `fin.get(c)`.
- Copia il carattere appena letto nell'array `text` e, se il carattere corrisponde alla prima lettera di una parola, assegna all'elemento corrente dell'array `pwords` l'indirizzo dell'elemento dell'array `text` in cui è appena stato copiato il carattere. Per semplicità, si assuma che un carattere rappresenta la prima lettera di una parola se esso è una lettera e se il carattere precedente è uno spazio o il carattere new line ('`\n`').
- Chiama una funzione `ContaParole` che riceve come parametro un array di puntatori a `char` e restituisce un intero che corrisponde al numero di parole trovate.
- Chiama inoltre una funzione `GeneraParole` che riceva come parametro un array di puntatori a `char`, un numero intero che è il valore di ritorno di `ContaParole`, e un array di 11 `char`. La funzione costruisce una parola di 10 `char` ottenuti estraendo in modo casuale 10 delle lettere iniziali puntate dai puntatori non nulli contenuti nell'array passato alla funzione. La funzione non ritorna alcun valore, ma popola correttamente l'array di 11 `char` (inserendo il terminatore di stringa nell'ultimo elemento).
- Al termine della lettura del file chiude il file e stampa a video (1) il testo letto in input (cioè il contenuto dell'array `text`), (2) il numero di parole trovate (cioè il numero di elementi di `pwords` diverso da `NULL`), (3) tutte le prime lettere delle parole trovate nel testo (cioè il carattere puntato da ogni elemento di `pwords` diverso da `NULL`), e (4) la stringa popolata da `GeneraParole`.

Il file `esercizio1.cpp` contiene gli include che definiscono le funzioni consentite ed è da completare con quanto richiesto sopra. Di seguito è riportato un esempio di esecuzione (la formattazione dell'output deve essere rispettata).

```
marco > ./a.out testo.txt
Text: Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura, ch  la diritta via era smarrita

Number of words: 19
Initials: Nmdcdnvmrpusocldves
Generated word: mlddpmpumv
```

Note:

- Scaricare il file `esercizio1.cpp`, modificarlo per inserire il codice necessario per rispondere a questo esercizio. **Caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.

- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`, `fstream`, `cstdlib`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta **NON** deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).

`esercizio1.cpp`

`testo.txt`

Information for graders:

Total of marks: 20