

Esame 20220223

Esercizio 1

(1) Esercizio 1 v1

ESSAY marked out of 10 penalty 0 File picker

Sia `input.txt` un file di input contenente stringhe separate da spazi. Queste stringhe possono essere ripetute molteplici volte all'interno del file.

Scrivere nel file `esercizio1.cc` un programma che, dati come argomenti da riga di comando il file di input e il nome di un file di output, scriva sul file in output tutte le stringhe **valide** senza duplicati. Una stringa é definita valida quando é composta solo da **lettere minuscole**.

Si scriva inoltre vicino ad ogni stringa il numero di occorrenze di quella stringa all'interno del file, seguendo la formattazione dell'esempio (`<stringa>: <numero>`). Il programma deve anche controllare l'accuratezza dell'invocazione (e.g., numero di argomenti corretti) e la corretta apertura degli stream di input e output.

Supponiamo che il primo file `input.txt` contenga

```
aaa AAA bfe csCcs bfe asas
fre aaa bfe pewdfs assda lsA
```

Il programma dovrà poter essere chiamato nel seguente modo:

```
./esercizio1.out input.txt output.txt
```

e dovrà produrre un file chiamato `output.txt` che conterrà i seguenti valori:

```
aaa: 2
bfe: 3
asas: 1
fre: 1
pewdfs: 1
assda: 1
```

Note:

- Si assuma inoltre che ogni stringa sia formata da al massimo 100 caratteri e che il numero di sequenze uniche sia al massimo 100.
- E' consentito l'utilizzo della funzioni `int strlen (const char * str), int strcmp (const char * str1, const char * str2)` e `char * strcpy (char * destination, const char * source)` della libreria `<cstring>`.
- Non é consentito l'utilizzo di altre funzioni di libreria "particolari" diverse da quelle specificate sopra o da quelle standard necessarie a risolvere l'esercizio.
- Le uniche assunzioni che si possono fare sull'input e su dimensioni di eventuali strutture/array utilizzate nel file di partenza fornito sono **solo quelle espressamente specificate in questo testo** (e NON quelle riportate nel file fornito, che sono SOLO indicative per consentire di svolgere l'esame).
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static`.

- Si ricorda che, l'esempio di esecuzione è puramente indicativo, 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.

Information for graders:

(2) Esercizio 1 v2

ESSAY

marked out of 10

penalty 0

File picker

Sia `input.txt` un file di input contenente stringhe separate da spazi. Queste stringhe possono essere ripetute molteplici volte all'interno del file.

Scrivere nel file `eserciziol.cc` un programma che, dati come argomenti da riga di comando il file di input e il nome di un file di output, scriva sul file in output tutte le stringhe **valide** senza duplicati. Una stringa é definita valida quando é composta solo da **lettere maiuscole**.

Si scriva inoltre vicino ad ogni stringa il numero di occorrenze di quella stringa all'interno del file, seguendo la formattazione dell'esempio (`<stringa>: <numero>`). Il programma deve anche controllare l'accuratezza dell'invocazione (e.g., numero di argomenti corretti) e la corretta apertura degli stream di input e output.

Supponiamo che il primo file `input.txt` contenga

```
AAA aaa BFE CScCS BFE ASAS
FRE AAA BFE PEWDFS ASSDA LSa
```

Il programma dovrà poter essere chiamato nel seguente modo:

```
./eserciziol.out input.txt output.txt
```

e dovrà produrre un file chiamato `output.txt` che conterrà i seguenti valori:

```
AAA: 2
BFE: 3
ASAS: 1
FRE: 1
PEWDFS: 1
ASSDA: 1
```

Note:

- Si assuma inoltre che ogni stringa sia formata da al massimo 100 caratteri e che il numero di sequenze uniche sia al massimo 100.
- E' consentito l'utilizzo della funzioni `int strlen (const char * str), int strcmp (const char * str1, const char * str2)` e `char * strcpy (char * destination, const char * source)` della libreria `<cstring>`.
- Non é consentito l'utilizzo di altre funzioni di libreria "particolari" diverse da quelle specificate sopra o da quelle standard necessarie a risolvere l'esercizio.
- Le uniche assunzioni che si possono fare sull'input e su dimensioni di eventuali strutture/array utilizzate nel file di partenza fornito sono **solo quelle espressamente specificate in questo testo** (e NON quelle riportate nel file fornito, che sono SOLO indicative per consentire di svolgere l'esame).
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static`.
- Si ricorda che, l'esempio di esecuzione è puramente indicativo, 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.

Information for graders:

(3) Esercizio 1 v3

ESSAY marked out of 10 penalty 0 File picker

Sia `input.txt` un file di input contenente stringhe separate da spazi. Queste stringhe possono essere ripetute molteplici volte all'interno del file.

Scrivere nel file `esercizio1.cc` un programma che, dati come argomenti da riga di comando il file di input e il nome di un file di output, scriva sul file in output tutte le stringhe **valide** senza duplicati. Una stringa é definita valida quando rappresenta un **numero dispari** (utilizzare la funzione `int atoi (const char * str)` di `cstdlib` per la conversione).

Si scriva inoltre vicino ad ogni stringa il numero di occorrenze di quella stringa all'interno del file, seguendo la formattazione dell'esempio (`<stringa>: <numero>`). Il programma deve anche controllare l'accuratezza dell'invocazione (e.g., numero di argomenti corretti) e la corretta apertura degli stream di input e output.

Supponiamo che il primo file `input.txt` contenga

```
793 231 16 793 80 9711 951 78 522
47 617 25 72 793 95 26 22 793 793
```

Il programma dovrà poter essere chiamato nel seguente modo:

```
./esercizio1.out input.txt output.txt
```

e dovrà produrre un file chiamato `output.txt` che conterrà i seguenti valori:

```
793: 5
231: 1
9711: 1
951: 1
47: 1
617: 1
25: 1
95: 1
```

Note:

- Si assuma inoltre che ogni stringa sia formata da al massimo 100 caratteri e che il numero di sequenze uniche sia al massimo 100.
- E' consentito l'utilizzo della funzioni `int strlen (const char * str)`, `int strcmp (const char * str1, const char * str2)` e `char * strcpy (char * destination, const char * source)` della libreria `<cstring>`.
- Non é consentito l'utilizzo di altre funzioni di libreria "particolari" diverse da quelle specificate sopra o da quelle standard necessarie a risolvere l'esercizio.
- Le uniche assunzioni che si possono fare sull'input e su dimensioni di eventuali strutture/array utilizzate nel file di partenza fornito sono **solo quelle espressamente specificate in questo testo** (e NON quelle riportate nel file fornito, che sono SOLO indicative per consentire di svolgere l'esame).
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static`.

- Si ricorda che, l'esempio di esecuzione è puramente indicativo, 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.

Information for graders:

(4) Esercizio 1 v4

ESSAY

marked out of 10

penalty 0

File picker

Sia `input.txt` un file di input contenente stringhe separate da spazi. Queste stringhe possono essere ripetute molteplici volte all'interno del file.

Scrivere nel file `esercizio1.cc` un programma che, dati come argomenti da riga di comando il file di input e il nome di un file di output, scriva sul file in output tutte le stringhe **valide** senza duplicati. Una stringa é definita valida quando rappresenta un **numero pari** (utilizzare la funzione `int atoi (const char * str)` di `cstdlib` per la conversione).

Si scriva inoltre vicino ad ogni stringa il numero di occorrenze di quella stringa all'interno del file, seguendo la formattazione dell'esempio (`<stringa>: <numero>`). Il programma deve anche controllare l'accuratezza dell'invocazione (e.g., numero di argomenti corretti) e la corretta apertura degli stream di input e output.

Supponiamo che il primo file `input.txt` contenga

```
600 248 80 2222 98 95 78 80
2222 98 25 248 95 2222 80
```

Il programma dovrà poter essere chiamato nel seguente modo:

```
./esercizio1.out input.txt output.txt
```

e dovrà produrre un file chiamato `output.txt` che conterrà i seguenti valori:

```
600: 1
248: 2
80: 3
2222: 3
98: 2
78: 1
52: 1
```

Note:

- Si assuma inoltre che ogni stringa sia formata da al massimo 100 caratteri e che il numero di sequenze uniche sia al massimo 100.
- E' consentito l'utilizzo della funzioni `int strlen (const char * str)`, `int strcmp (const char * str1, const char * str2)` e `char * strcpy (char * destination, const char * source)` della libreria `<cstring>`.
- Non é consentito l'utilizzo di altre funzioni di libreria "particolari" diverse da quelle specificate sopra o da quelle standard necessarie a risolvere l'esercizio.
- Le uniche assunzioni che si possono fare sull'input e su dimensioni di eventuali strutture/array utilizzate nel file di partenza fornito sono **solo quelle espressamente specificate in questo testo** (e NON quelle riportate nel file fornito, che sono SOLO indicative per consentire di svolgere l'esame).
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static`.

- Si ricorda che, l'esempio di esecuzione è puramente indicativo, 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.

Information for graders:

Total of marks: 40