

Esame 20220729

Esercizio 1

(1) Esercizio 1 v1

ESSAY marked out of 10 penalty 0 File picker

Scrivere nel file `esercizio1.cc` programma che, presi come argomento del main:

- Un file contenente una lista non ordinata di voti compresi tra 18 e 30.
- Un intero `DIM` che rappresenta il numero totale di elementi presenti nel file di input.
- Un intero `N`, minore o uguale a `DIM`.

crei un nuovo file di testo (chiamato `output.txt`) contenente una lista ordinata con gli `N` voti più alti presenti nel file passato da riga di comando.

Supponiamo che il primo file `input.txt` contenga

```
23
30
18
27
20
21
18
25
```

allora il comando

```
./esercizio1.out input.txt 8 3
```

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

```
30
27
25
```

Note:

- Utilizzare la funzione `sort_array()` fornita per effettuare l'ordinamento.
- Non é concesso fare assunzioni sul numero **massimo** di elementi presenti nel file pena l'annullamento dell'esercizio.
- É consentito l'utilizzo della funzione `int atoi (const char * str)`.
- 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.
- Non é consentito inizializzare array dinamici con la sintassi `int array[n]`, dove `n` é una variabile non definita a compile time.
- 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

Scrivere nel file `esercizi01.cc` programma che, presi come argomento del main:

- Un file contenente una lista non ordinata di voti compresi tra 18 e 30.
- Un intero `DIM` che rappresenta il numero totale di elementi presenti nel file di input.
- Un interno `N`, minore o uguale a `DIM`.

crei un nuovo file di testo (chiamato `output.txt`) contenente una lista ordinata con gli `N` voti piú bassi presenti nel file passato da riga di comando.

Supponiamo che il primo file `input.txt` contenga

```
23
30
18
27
20
21
18
25
```

allora il comando

```
./esercizi01.out input.txt 8 4
```

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

```
18
18
20
21
```

Note:

- Utilizzare la funzione `sort_array()` fornita per effettuare l'ordinamento.
- Non é concesso fare assunzioni sul numero **massimo** di elementi presenti nel file pena l'annullamento dell'esercizio.
- É consentito l'utilizzo della funzione `int atoi (const char * str)`.
- 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.
- Non é consentito inizializzare array dinamici con la sintassi `int array[n]`, dove `n` é una variabile non definita a compile time.
- 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

Scrivere nel file `eserciziol.cc` programma che, presi come argomento del main:

- Un file contenente una lista non ordinata di caratteri.
- Un intero `DIM` che rappresenta il numero totale di elementi presenti nel file di input.
- Un interno `N`, minore o uguale a `DIM`.

crei un nuovo file di testo (chiamato `output.txt`) contenente una lista ordinata con gli `N` caratteri piú alti presenti nel file passato da riga di comando. Si assuma che il carattere `a` sia piú alto di `b`, che `b` sia piú alto di `c` e cosí via.

Supponiamo che il primo file `input.txt` contenga

a
z
c
s
w
b
d

allora il comando

```
./eserciziol.out input.txt 7 3
```

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

a
b
c

Note:

- Utilizzare la funzione `sort_array()` fornita per effettuare l'ordinamento.
- Non é concesso fare assunzioni sul numero **massimo** di elementi presenti nel file pena l'annullamento dell'esercizio.
- 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.
- Non é consentito inizializzare array dinamici con la sintassi `int array[n]`, dove `n` é una variabile non definita a compile time.
- 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

Scrivere nel file `esercizio1.cc` programma che, presi come argomento del main:

- Un file contenente una lista non ordinata di caratteri.
- Un intero `DIM` che rappresenta il numero totale di elementi presenti nel file di input.
- Un intero `N`, minore o uguale a `DIM`.

crei un nuovo file di testo (chiamato `output.txt`) contenente una lista ordinata con gli `N` caratteri piú alti presenti nel file passato da riga di comando. Si assuma che il carattere `z` sia piú basso di `y`, che `y` sia piú alto di `x` e cosí via.

Supponiamo che il primo file `input.txt` contenga

```
a
z
c
s
x
y
b
d
```

allora il comando

```
./esercizio1.out input.txt 8 3
```

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

```
z
y
x
```

Note:

- Utilizzare la funzione `sort_array()` fornita per effettuare l'ordinamento.
- Non é concesso fare assunzioni sul numero **massimo** di elementi presenti nel file pena l'annullamento dell'esercizio.
- 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.
- Non é consentito inizializzare array dinamici con la sintassi `int array[n]`, dove `n` é una variabile non definita a compile time.
- 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