

Esercizio 1

(1) Esercizio 1 v1

ESSAY marked out of 10 penalty 0 File picker

Un'azienda che gestisce impianti di risalita sciistici emette una serie di abbonamenti per i propri clienti. Allo scopo di elaborare le informazioni riguardanti tali abbonamenti si sviluppi, in linguaggio C++, un programma che opera come segue:

- Riceva come parametro da linea di comando il nome del file che contiene le informazioni degli abbonamenti emessi nell'anno. Si controlli che il numero di parametri sia uno:
- Controlli se il file esiste, in caso non esista segnali un errore e termini il programma.
- Nel caso il file esista, legga la prima riga che contiene il numero di abbonamenti emessi nell'anno e allochi un array `abb` di `N` strutture `Abbonamento`. Una struttura `Abbonamento` ha i seguenti campi: il numero dell'abbonamento (un numero intero), il nome e il cognome dell'abbonato (due stringhe di 31 caratteri più il terminatore), il tipo di abbonamento (un carattere: 'a' per stagionale, 'm' per mensile, 's' per settimanale e 'g' per giornaliero), il costo dell'abbonamento (un numero reale). Tale costo è pari a 10 euro per il giornaliero, 50 euro per il settimanale, 150 euro per il mensile, 500 euro per lo stagionale.
- Legga quindi le successive `N` righe, ciascuna delle quali contiene le sovramentionate informazioni relative a un abbonamento (escluso il costo, da aggiungere secondo le regole sopra descritte), e popoli l'array `abb`.
- Chiuda il file al termine della lettura.
- Chiami una funzione `guadagno` che riceva in ingresso, ovvero come parametro, un array di strutture `Abbonamento`, la sua dimensione `n` e calcoli e restituisca come valore di ritorno il guadagno totale degli abbonamenti emessi, sapendo che le spese sostenute dall'azienda sono quantificabili come il 10% del costo di ogni abbonamento.
- Stampi il guadagno totale ottenuto dall'azienda nell'anno.
- Chiami una funzione `elabora` che riceve in ingresso, ovvero come parametro, un array di strutture `Abbonamento`, la sua dimensione `n` e un array `r` di 4 numeri reali. Dopo avere inizializzato l'array `r` ponendo tutti i suoi elementi a 0, la funzione calcolerà e inserirà nel primo elemento di `r` il ricavo totale ottenuto dagli abbonamenti stagionali, nel secondo elemento di `r` il ricavo totale ottenuto dagli abbonamenti mensili, nel terzo elemento di `r` il ricavo totale ottenuto dagli abbonamenti settimanali e nel quarto elemento di `r` il ricavo totale ottenuto dagli abbonamenti giornalieri. La funzione restituisce come valore di ritorno il carattere corrispondente alla tipologia di abbonamento ('a', 'm', 's' o 'g') che ha realizzato il massimo ricavo.
- Apra in scrittura il file `risultati.txt` e scriva nel file il numero dell'abbonamento, il nome e il cognome dell'abbonato per tutti gli abbonamenti contenuti nell'array `abb` della tipologia che ha ottenuto il massimo ricavo, come riportato dal valore di ritorno di `elabora`. Ad esempio, se dal valore di ritorno di `elabora` si evince che gli abbonamenti mensili hanno ottenuto il massimo ricavo, il programma stamperà nel file `risultati.txt` le informazioni relative ai soli abbonamenti mensili contenuti in `abb`.

Di seguito è riportato un esempio di esecuzione usando i file in allegato.

```
marco > ./a.out abbonamenti.txt && cat risultati.txt
Guadagno totale: 1278
4 Martina Battisti m 150
5 Dante Alighieri m 150
9 Matteo Salvini m 150
16 Donald Trump m 150
marco > ./a.out abbonamenti1.txt && cat risultati.txt
Guadagno totale: 1188
1 Marco Roveri a 500
marco > ./a.out abbonamenti2.txt && cat risultati.txt
Guadagno totale: 693
1 Marco Roveri s 50
2 Giovanna Varni s 50
3 Andrea Naimoli s 50
4 Martina Battisti s 50
5 Dante Alighieri s 50
6 Giovanni Pascoli s 50
7 Michele Santoro s 50
13 Paperino Paolino s 50
```

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`

`abbonamenti.txt`

`abbonamenti1.txt`

`abbonamenti2.txt`

Information for graders:

(2) Esercizio 2 v1

ESSAY

marked out of 10

penalty 0

File picker

Un'azienda che gestisce impianti di risalita sciistici emette una serie di abbonamenti per i propri clienti. Allo scopo di elaborare le informazioni riguardanti tali abbonamenti si sviluppi, in linguaggio C++, un programma che opera come segue:

- Riceva come parametro da linea di comando il nome del file che contiene le informazioni degli abbonamenti emessi nell'anno. Si controlli che il numero di parametri sia uno:
- Controlli se il file esiste, in caso non esista segnali un errore e termini il programma.
- Nel caso il file esista, legga la prima riga che contiene il numero di abbonamenti emessi nell'anno e allochi un array `abb` di `N` strutture `Abbonamento`. Una struttura `Abbonamento` ha i seguenti campi: il `numero` dell'abbonamento (un numero intero), il `nome` e il `cognome` dell'abbonato (due stringhe di 31 caratteri più il terminatore), il `tipo` di abbonamento (un carattere: `'a'` per stagionale, `'m'` per mensile, `'s'` per settimanale e `'g'` per giornaliero), il `costo` dell'abbonamento (un numero reale). Tale costo è pari a 10 euro per il giornaliero, 50 euro per il settimanale, 150 euro per il mensile, 500 euro per lo stagionale.
- Legga quindi le successive `N` righe, ciascuna delle quali contiene le sovramenzionate informazioni relative a un abbonamento (escluso il costo, da aggiungere secondo le regole sopra descritte), e popoli l'array `abb`.
- Chiuda il file al termine della lettura.
- Chiami una funzione `ricavo` che riceva in ingresso, ovvero come parametro, un array di strutture `Abbonamento`, la sua dimensione `n` e calcoli e restituisca come valore di ritorno il ricavo totale degli abbonamenti emessi, sapendo che le spese sostenute dall'azienda sono quantificabili come il 10% del costo di ogni abbonamento.
- Stampi il ricavo totale ottenuto dall'azienda nell'anno.
- Chiami una funzione `elabora` che riceve in ingresso, ovvero come parametro, un array di strutture `Abbonamento`, la sua dimensione `n` e un array `r` di 4 numeri reali. Dopo avere inizializzato l'array `r` ponendo tutti i suoi elementi a 0, la funzione calcolerà e inserirà nel primo elemento di `r` il guadagno totale ottenuto dagli abbonamenti stagionali, nel secondo elemento di `r` il guadagno totale ottenuto dagli abbonamenti mensili, nel terzo elemento di `r` il guadagno totale ottenuto dagli abbonamenti settimanali e nel quarto elemento di `r` il guadagno totale ottenuto dagli abbonamenti giornalieri. La funzione restituisce come valore di ritorno il carattere corrispondente alla tipologia di abbonamento (`'a'`, `'m'`, `'s'` o `'g'`) che ha realizzato il massimo guadagno.
- Apra in scrittura il file `risultati.txt` e scriva nel file il numero dell'abbonamento, il nome e il cognome dell'abbonato per tutti gli abbonamenti contenuti nell'array `abb` della tipologia che ha ottenuto il massimo guadagno, come riportato dal valore di ritorno di `elabora`. Ad esempio, se dal valore di ritorno di `elabora` si evince che gli abbonamenti mensili hanno ottenuto il massimo guadagno, il programma stamperà nel file `risultati.txt` le informazioni relative ai soli abbonamenti mensili contenuti in `abb`.

Di seguito è riportato un esempio di esecuzione usando i file in allegato.

```
marco > ./a.out abbonamenti.txt && cat risultati.txt
Ricavo totale: 1420
4 Martina Battisti m 150
5 Dante Alighieri m 150
9 Matteo Salvini m 150
16 Donald Trump m 150
```

```
marco > ./a.out abbonamenti1.txt && cat risultati.txt
Ricavo totale: 1320
1 Marco Roveri a 500
marco > ./a.out abbonamenti2.txt && cat risultati.txt
Ricavo totale: 770
1 Marco Roveri s 50
2 Giovanna Varni s 50
3 Andrea Naimoli s 50
4 Martina Battisti s 50
5 Dante Alighieri s 50
6 Giovanni Pascoli s 50
7 Michele Santoro s 50
13 Paperino Paolino s 50
```

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`

`abbonamenti.txt`

`abbonamenti1.txt`

`abbonamenti2.txt`

Information for graders:

Total of marks: 20