

Esame 20250221

Esercizio 1

(1) Esercizio 1 v1

Un sistema informatico universitario ha un modulo per la gestione dei quiz a risposta multipla. Tale modulo funziona nel seguente modo: ogni volta che viene somministrato un quiz, il sistema genera un file `Soluzioni.txt` che contiene una sola colonna (non possono verificarsi altri casi) con le soluzioni delle domande del quiz. La soluzione a ogni domanda è identificata da un numero intero tra 1 e 4. Il quiz è composto da 10 domande. Inoltre, il sistema genera per ogni studente un file, sempre con una sola colonna, contenente le risposte che sono numeri interi da 0 a 4, 0 indicante l'opzione "Preferisco non rispondere". Il file ha nome `StudenteX.txt`, ove X è un intero progressivo che parte da 1 e che identifica lo studente.

Si sviluppi quanto segue:

- (a) La funzione `leggi_file` che riceve come parametri il nome di un file (una stringa), un array di interi e la sua dimensione. La funzione apra il file il cui nome è passato come parametro, verificando che esista. Se non esiste, la funzione restituisce -1. La funzione riempie come prima cosa l'array di zeri e, successivamente, vi copia i valori contenuti nel file. La funzione restituisce il numero di valori copiati.
- (b) La funzione `check` che riceva come parametro un array di interi, la sua dimensione e due ulteriori interi che rappresentano gli estremi inferiore e superiore di un intervallo. La funzione verifica che il contenuto dell'array sia valido, ovvero che i valori siano compresi nell'intervallo suddetto. La funzione restituisce 0 in caso di controllo fallito, 1 in caso di controllo completato con successo.
- (c) La funzione `voto` che prende come parametri due array di interi (della stessa dimensione) e la loro dimensione (una sola volta) e confronti il contenuto dei due array. La funzione calcola il voto scandendo gli array elemento per elemento: ogni volta che trova due valori uguali, somma 3 punti al voto, ogni volta che trova due valori diversi, se il valore nel secondo array non è 0, sottrae 1 al voto, altrimenti prosegue nella scansione.

Si scriva un programma C++ che:

- Definisca un array `sol` di 10 numeri interi;
- Chiami la funzione `leggi_file` passando come parametro la stringa "`Soluzioni.txt`", `sol` e la sua dimensione; nel caso in cui la funzione torni -1 il programma termini con un messaggio di errore;
- Chiami la funzione `check` che riceva come parametro l'array `sol` e la sua dimensione, e i valori 1 e 4 che costituiscono gli estremi dell'intervallo di interesse. Il programma termina con un messaggio di errore in caso il controllo fallisca.
- Chieda all'utente di inserire il numero di studenti partecipanti all'appello (attenzione: il numero dei partecipanti corrisponde al più grande X);
- Per ciascuno studente *i* (da 1 a X):
 - definisca una stringa `nome` di dimensione 32 inizializzata a "Studente", e un array `risp` di 10 numeri interi;
 - chiami la funzione `leggi_file` passando la stringa "`Studentei.txt`" dove *i* è l'identificativo dello studente *i*-esimo, un array `risp` atto a contenere le risposte dello studente e la sua dimensione. Se uno dei file non esiste, il programma segnala un errore e procede con i file successivi;

- chiami quindi la funzione `voto` per confrontare il contenuto di `s` e `r`. `r` sia passato come secondo array.
- Scriva in fondo al file "Studente*i*.txt" il voto dello studente, oppure "Insuff" se il voto è inferiore a 18 e poi chiuda il file. Il contenuto precedente del file "Studente*i*.txt" deve essere preservato.
- Termini.

Non è consentito la definizione o uso di altre funzioni rispetto a quelle richieste o indicate in questo testo.

Un esempio di esecuzione del programma è il seguente:

```
computer > ./a.out
Inserisci il numero degli studenti partecipanti a questo appello: 5
Sto elaborando il file: Studente1.txt
Sto elaborando il file: Studente2.txt
Errore nell'apertura del file!
Sto elaborando il file: Studente3.txt
Sto elaborando il file: Studente4.txt
Sto elaborando il file: Studente5.txt
Errore nell'apertura del file!
```

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.
- Per convertire un intero `n` in una stringa (array di `char`) si usi la seguente linea di codice: `to_string(n).c_str()`.
- Per concatenare due stringhe (array di `char`) si usi la funzione `strcat` della libreria `cstring`.
- 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`, `cstring`.
- 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 Soluzioni.txt Studente1.txt Studente3.txt Studente4.txt

Information for graders:

(2) Esercizio 1 v2

Un sistema informatico universitario ha un modulo per la gestione dei quiz a risposta multipla. Tale modulo funziona nel seguente modo: ogni volta che viene somministrato un quiz, il sistema genera un file Soluzioni.txt che contiene una sola colonna (non possono verificarsi altri casi) con le soluzioni delle domande del quiz. La soluzione a ogni domanda è identificata da un numero intero tra 1 e 3. Il quiz è composto da 15 domande. Inoltre, il sistema genera per ogni studente un file, sempre con una sola colonna, contenente le risposte che sono numeri interi da 0 a 3, 0 indicante l'opzione "Preferisco non rispondere". Il file ha nome StudenteX.txt, ove X è un intero progressivo che parte da 1 e che identifica lo studente.

Si sviluppi quanto segue:

- (a) La funzione `leggi` che riceve come parametri il nome di un file (una stringa), un array di interi e la sua dimensione. La funzione apra il file il cui nome è passato come parametro, verificando che esista. Se non esiste, la funzione restituisce -1. La funzione riempie come prima cosa l'array di zeri e, successivamente, vi copia i valori contenuti nel file. La funzione restituisce il numero di valori copiati.
- (b) La funzione `verifica` che riceva come parametro un array di interi, la sua dimensione e due ulteriori interi che rappresentano gli estremi inferiore e superiore di un intervallo. La funzione verifica che il contenuto dell'array sia valido, ovvero che i valori siano compresi nell'intervallo suddetto. La funzione restituisce 0 in caso di controllo fallito, 1 in caso di controllo completato con successo.
- (c) La funzione `voto` che prende come parametri due array di interi (della stessa dimensione) e la loro dimensione (una sola volta) e confronti il contenuto dei due array. La funzione calcola il voto scandendo gli array elemento per elemento: ogni volta che trova due valori uguali, somma 2 punti al voto, ogni volta che trova due valori diversi, se il valore nel secondo array non è 0, sottrae 1 al voto, altrimenti prosegue nella scansione.

Si scriva un programma C++ che:

- Definisca un array `sol` di 15 numeri interi;
- Chiami la funzione `leggi` passando come parametro la stringa "Soluzioni.txt", `sol` e la sua dimensione; nel caso in cui la funzione torni -1 il programma termini con un messaggio di errore;
- Chiami la funzione `verifica` che riceva come parametro l'array `sol` e la sua dimensione, e i valori 1 e 3 che costituiscono gli estremi dell'intervallo di interesse. Il programma termina con un messaggio di errore in caso il controllo fallisca.
- Chieda all'utente di inserire il numero di studenti partecipanti all'appello (attenzione: il numero dei partecipanti corrisponde al più grande X);
- Per ciascuno studente i (da 1 a X):
 - definisca una stringa `nome` di dimensione 24 inizializzata a "Studente", e un array `risp` di 15 numeri interi;
 - chiami la funzione `leggi` passando la stringa "Studentei.txt" dove i è l'identificativo dello studente i -esimo, un array `risp` atto a contenere le risposte dello studente e la sua dimensione. Se uno dei file non esiste, il programma segnala un errore e procede con i file successivi;
 - chiami quindi la funzione `voto` per confrontare il contenuto di `sol` e `risp`. `risp` sia passato come secondo array.

- Scriva in fondo al file "Studentei.txt" il voto dello studente, oppure "Non sufficiente" se il voto è inferiore a 18 e poi chiuda il file. Il contenuto precedente del file "Studentei.txt" deve essere preservato.
- Termini.

Non è consentito la definizione o uso di altre funzioni rispetto a quelle richieste o indicate in questo testo.

Un esempio di esecuzione del programma è il seguente:

```
computer > ./a.out
Inserisci il numero degli studenti partecipanti a questo appello: 5
Sto elaborando il file: Studente1.txt
Sto elaborando il file: Studente2.txt
Errore nell'apertura del file!
Sto elaborando il file: Studente3.txt
Sto elaborando il file: Studente4.txt
Sto elaborando il file: Studente5.txt
Errore nell'apertura del file!
```

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.
- Per convertire un intero `n` in una stringa (array di `char`) si usi la seguente linea di codice: `to_string(n).c_str()`.
- Per concatenare due stringhe (array di `char`) si usi la funzione `strcat` della libreria `cstring`.
- 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`, `cstring`.
- 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 Soluzioni.txt Studente1.txt Studente3.txt Studente4.txt

Information for graders:

Total of marks: 20