

Istruzioni

- Tempo disponibile: 120 minuti.
- Non è permesso l'uso di dispositivi elettronici (a parte il PC della propria postazione).
- Il programma sarà valutato per
 - Identificazione delle strutture dati e degli algoritmi appropriati alle specifiche
 - Corretta implementazione di strutture dati e algoritmi
 - Utilizzo efficiente delle risorse
 - Stile (chiarezza, utilizzo di costrutti appropriati, corretta strutturazione)
- I programmi non compilabili saranno valutati 0 punti.
- Fare l'upload di tutti i file che compongono il programma.
- Visual Studio Code è installato su tutte le postazioni. Si può comunque usare l'editor che si preferisce fra quelli installati.

Esercizio - Parte 1 (max 16 punti)

In un file binario è memorizzata la cronologia web (semplificata) di un browser. Per ogni visita di una pagina web, è presente un record composto da

- nome del sito visitato (array di 30 `char`, compreso il terminatore);
- anno (`int`);
- mese (`int`);
- giorno (`int`).

Ad esempio, il file allegato `visite.dat` contiene i dati mostrati in Tabella 1.

Si scriva un programma in linguaggio C che produca l'elenco dei siti visitati con relative date e, in particolare,

- riceva come argomento della linea di comando il nome di un file binario del formato sopra indicato;
- crei una lista collegata con un elemento per ogni combinazione di sito e data trovata nel file, ordinata
 - per data crescente

– a parità di data, alfabeticamente per sito

e senza ripetizioni (cioè se nella stessa data lo stesso sito è stato visitato più volte, nella lista deve essere presente un solo nodo per quella combinazione di data e sito);

- crei un file di testo di nome `elenco.txt` con una riga per ogni nodo della lista, contenente data e sito.

Ad esempio, se l'eseguibile prodotto si chiama `cronologia` e `visite.dat` è il file allegato, l'invocazione `./cronologia visite.dat` deve produrre un file `elenco.txt` di contenuto simile al seguente:

```
16/9/2019 - amazon.it
16/9/2019 - ebay.it
16/9/2019 - google.it
16/9/2019 - libero.it
16/9/2019 - youtube.com
17/9/2019 - libero.it
17/9/2019 - repubblica.it
17/9/2019 - wikipedia.org
17/9/2019 - yahoo.com
18/9/2019 - amazon.it
18/9/2019 - facebook.com
18/9/2019 - google.com
18/9/2019 - google.it
18/9/2019 - repubblica.it
18/9/2019 - youtube.com
19/9/2019 - amazon.it
19/9/2019 - google.com
19/9/2019 - google.it
19/9/2019 - repubblica.it
19/9/2019 - yahoo.com
19/9/2019 - youtube.com
```

Ulteriori specifiche

- La lista collegata deve essere definita come tipo di dato astratto, cioè separando interfaccia e implementazione.
- Verificare la correttezza della linea di comando e la corretta apertura dei file; in caso di errore, stampare un messaggio e terminare l'esecuzione.
- Il programma deve essere costituito dai seguenti file:
 - `main.c` contenente (tra eventuali altre) la funzione `main`;

- `listaVisite.c` con la definizione delle funzioni su liste (ed eventuali altre);
- `listaVisite.h` con le definizioni dei tipi di dato e le dichiarazioni delle funzioni definite in `listaVisite.c` e utilizzate in `main.c`;
- `Makefile` che permetta di costruire l'eseguibile con un singolo comando `make`.

Esercizio - Parte 2 (max 6 punti)

Estendere il programma sviluppato per la parte 1 in modo che, se riceve come secondo argomento della linea di comando una stringa rappresentante un intero n , prima di stampare l'elenco dei siti visitati elimini dalla cronologia i dati relativi agli ultimi n giorni, compreso il giorno dell'invocazione del programma.

Si ricorda che la funzione `sscanf` consente di leggere i dati formattati da una stringa, anziché da file. Ad esempio, data la definizione `int a;`, l'istruzione `sscanf("3", "%d", &a);` assegna ad `a` il valore intero 3.

Possono essere utili le funzioni definite nell'allegato `date.c` e dichiarate nell'allegato `date.h` di prototipo:

- `Data oggi()` che restituisce la data del giorno della chiamata;
- `int giorno_giuliano(Data d)` che restituisce il numero di giorni passati dall'1 gennaio 4713 a.C alla data passata come parametro.

Il tipo di dato `Data` è definito in `date.h`.

In caso si usino queste funzioni, è obbligatorio usare i file che le contengono come file del progetto, anziché copiare e incollare la definizione.

Sito	Anno	Mese	Giorno
amazon.it	2019	9	18
yahoo.com	2019	9	19
amazon.it	2019	9	19
libero.it	2019	9	16
repubblica.it	2019	9	17
google.it	2019	9	19
google.com	2019	9	19
amazon.it	2019	9	18
google.com	2019	9	18
google.it	2019	9	16
youtube.com	2019	9	16
yahoo.com	2019	9	17
google.it	2019	9	18
google.it	2019	9	19
yahoo.com	2019	9	19
repubblica.it	2019	9	18
google.it	2019	9	18
repubblica.it	2019	9	19
amazon.it	2019	9	16
youtube.com	2019	9	18
repubblica.it	2019	9	17
youtube.com	2019	9	19
wikipedia.org	2019	9	17
ebay.it	2019	9	16
facebook.com	2019	9	18
amazon.it	2019	9	18
youtube.com	2019	9	16
libero.it	2019	9	17
google.it	2019	9	18
libero.it	2019	9	17

Tabella 1: Contenuto dell'allegato [visite.dat](#)