



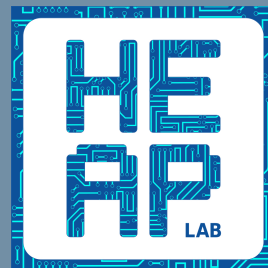
POLITECNICO
MILANO 1863

INFORMATICA (PER AEROSPAZIALI)

A.A. 2018-19

Laboratorio n°4

Dott. Michele Zanella



- I file rappresentano una sequenza di byte, siano essi *file di testo* o *file binari*.
- La fine del file è espressa dal carattere speciale **EOF**.
- Vengono rappresentati tramite una struttura **FILE**
- Nella manipolazione dei file si utilizza il puntatore a tale struttura: `FILE* fp;`
- La libreria `<stdlib.h>` mette a disposizione una serie di funzioni per operare sui file:
 - `FILE* fopen(const char* filename, const char* mode):` crea o apre un file, specificando il nome del file e la modalità di apertura.
 - **r**: lettura
 - **w**: scrittura
 - **a**: writing appending mode
 - `int fputc(int c, FILE* fp):` scrive il carattere `c` nel file puntato da `fp` tramite un *output stream*
 - `int fputs(const char* s, FILE *fp):` scrive una stringa `s` nel file puntato da `fp`
 - Altre simili a quelle di console: `fwrite`, `fprintf`...

- Altre funzioni
 - `int fgetc(FILE *fp)`: leggere un carattere dal file puntato da *fp*
 - `char* fgets(char* buf, int n, FILE *fp)`: legge fino a *n-1* caratteri dal file puntato da *fp*. La funzione si ferma se incontra un `'\n'` o la fine del file.
 - Alte simili a quelle di console: `fread`, `fscanf`...
 - `int fclose(FILE *fp)`: Chiude un file ed esegue un flush sui buffer di scrittura/lettura.
- La sequenza per operare sui file è:
 1. `fopen(...)`
 2. `fread/fgets` oppure `fwrite/fputs`
 3. `fclose`
- In caso di problemi in apertura il puntatore *fp* = NULL

Elaborazione delle immagini

Procedimento con il quale data un'immagine si eseguono delle operazioni per modificarne l'aspetto.

https://it.wikipedia.org/wiki/Elaborazione_digitale_delle_immagini

Immagine: file contenente delle informazioni circa la dimensione, i singoli pixel (colore, luminosità, posizione, ecc...)

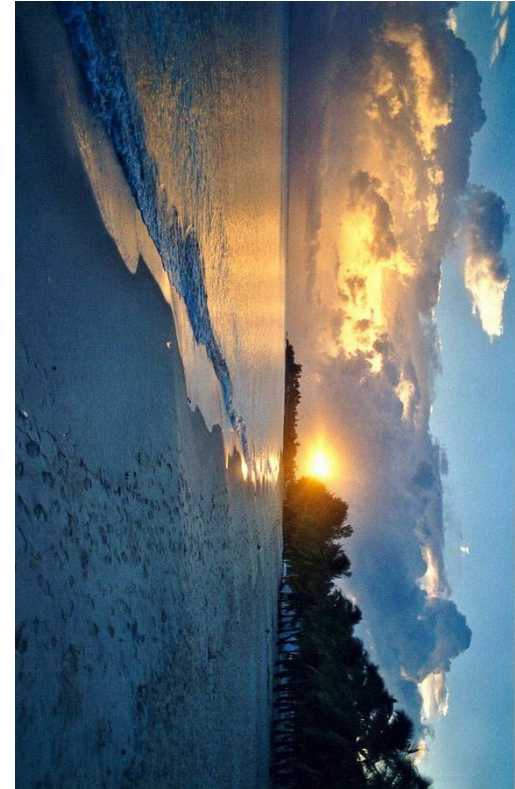
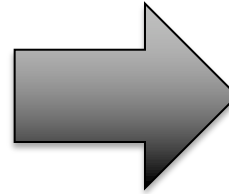
Può essere vista come una matrice alla quale si applicano delle funzioni:

- Traslazione
- Rotazione
- Inversione
- Riflessione
-

Fino ad operazioni più complesse come: classificazione, edge detection, regressione, pattern recognition...

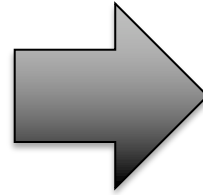
Elaborazione delle immagini (cont'd)

Rotazione



Elaborazione delle immagini (cont'd)

Negativo



Elaborazione delle immagini: PBM

Portable BitMap (PBM): è un formato immagine non compresso contenente solo informazioni sul colore monocromatico (bianco o nero) e la posizione dei singoli pixel.

<http://netpbm.sourceforge.net/doc/pbm.html>

Elaborazione delle immagini: PBM (cont'd)

Ogni file di immagine plain PBM è formato da due parti con alcune sezioni:

- **HEADER:**
 - "Magic Number": formato da due caratteri che identificano il tipo di file ("P1")
 - Whitespace (spazio bianco)
 - Larghezza dell'immagine: formattata in ASCII decimale
 - Whitespace
 - Altezza dell'immagine
 - Newline (a capo)
 - Eventuali commenti preceduti dal simbolo '#'
- **DATA:**
 - Una matrice di pixel: ciascun byte rappresenta un pixel, 1 se nero, 0 se bianco (in ASCII). Ciascuna linea è lunga al massimo 70 caratteri

Elaborazione delle immagini: PBM (cont'd)

Esempio: feep.pbm

Magic Number

P1

Commento

feep.pbm

Larghezza e Altezza

24 7

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	1	1	1	1	0	0	1	1	1	1	0	0	1	1	1	1	0	0	1	1	1	1	0
0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0
0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	1	0
0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	1	1	1	1	0	0	1	1	1	1	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Matrice di Pixel

Essendo un formato di immagine è apribile con un visualizzatore di immagini!

Esercizio 4.1: File.txt

Si implementi un programma che scriva più linee inserite dall'utente nel file *test.txt* e ne successivamente ne legga il contenuto.

- Attenzione ad aprire e chiudere il file nelle due modalità!

Esercizio 4.2: Data.csv

Si implementi un programma che legga dal file *data.csv* lo schedule dei voli di una compagnia.

Ogni volo è contraddistinto da: *codice del volo, codice IATA dell'aeroporto di partenza, codice IATA dell'aeroporto di arrivo, codice ICAO dell'aeromobile utilizzato e durata (in minuti) del volo.*

Infine chieda all'utente di aggiungere gli stessi dati per un nuovo volo.

- Dichiarare le struct, typedef e define necessarie
- Prototipare le funzioni necessarie
- Utilizzare la funzione `strtok` per parsare le linee del file csv.

Esercizio 4.3: Salvataggio partite Mastermind con File I/O

Aggiungere al gioco Mastermind la possibilità di salvare la partita e riprendere il gioco.

Requisiti:

- Il gioco deve salvare in un file di testo (. t x t) ogni round di una partita
- L'utente sceglie se riprendere l'esecuzione di una partita o se iniziarne una nuova
- Il gioco deve poter riprendere l'esecuzione di una partita dall'ultimo round giocato salvato su un file di testo (. t x t)

Hints:

- Se la partita termina (vittoria o sconfitta) svuotare il file di testo
- Aprire e CHIUDERE il file nelle modalità corrette!
- Se si inizia una partita nuova, sovrascrivere il file precedente
- Salvo solo l'ultimo round giocato (ulteriore esercizio: salvare tutta la partita fino all'ultimo round giocato)

Esercizio 4.4: Elaborazione di un'immagine PBM

Scrivere un programma che permetta di eseguire delle semplici elaborazioni di un'immagine in formato PBM.

Requisiti:

- Il programma deve caricare e leggere un'immagine da file in formato PBM
- Deve implementare la rotazione di 90° dell'immagine
- Deve mostrare il negativo dell'immagine
- Per ogni trasformazione deve salvare un nuovo file immagine in formato plain PBM

Hints:

- Salvare l'immagine caricata in una matrice e lavorare sulla matrice
- Scaricare lo zip con l'immagine "feep.pbm" dal sito del laboratorio ed estrarla nella stessa cartella dove andrete ad implementare il codice

Esercizio 4.4: Elaborazione di un'immagine PBM (cont'd)

Opzioni avanzate:

- Possibilità di passare come argomenti da command line le operazioni da eseguire sull'immagine
- Possibilità di passare come primo argomento il nome del file (senza estensione) su cui operare

Esercizio 4.4: Elaborazione di un'immagine PBM (cont'd)

Modello e tipi di dato:

Immagine: -> *struct*

- Tipo (i.e., il magic number) -> *char[2]*
- Larghezza -> *int*
- Altezza -> *int*
- Pixel -> *int[ALTEZZA][LARGHEZZA]*

Esercizio 4.4: Elaborazione di un'immagine PBM (cont'd)

Organizzazione delle funzioni:

- `apri_immagine`: Funzione che apre l'immagine, legge l'header e legge i dati dei pixel (suddivisa in sotto funzioni specifiche)
- `salva_immagine`: Funzione che scrive l'header, scrive i dati dei pixel e salva l'immagine (suddivisa in sotto funzioni specifiche)
- `copia_immagine`: Funzione che crea una copia dell'immagine originale per effettuare le operazioni
- `ruota_immagine_90`: Funzione che ruota l'immagine originale di 90 gradi
- `negativo_immagine`: Funzione che crea il negativo dell'immagine originale