

# TECNICHE DI PROGRAMMAZIONE, A.A. 2021/2022

## Esercitazione di Laboratorio 2

---

Valutazione: gli **esercizi 1 e 3** saranno oggetto di valutazione.

Scadenza: caricamento di quanto valutato - **entro le 23:59 dell'8/4/2022**. Andranno caricati insieme i laboratori 1, 2 e 3, con modalità indicate in un documento a parte.

### Obiettivi

- Risolvere problemi gestendo problemi iterativi su dati scalari (*Dal problema al programma: Cap. 3*), di tipo numerico (3.1) e di codifica (3.2)

### Contenuti tecnici

- Basi di Input Output
- Utilizzo di funzioni
- Costrutti condizionali e iterativi
- Manipolazioni elementari di numeri (int e float) e caratteri (char)

---

Da risolvere durante il laboratorio oppure prima/dopo il laboratorio stesso

### **Esercizio 1.**

*Competenze: IO formattato, manipolazioni di numeri;*

*Categoria: problemi numerici (Dal problema al programma: 3.1), con affinità ai problemi di codifica di numeri (3.2.1)*

Scrivere un programma C che, acquisiti 2 numeri interi positivi ne calcoli il massimo comune divisore utilizzando la formula di Eulero.

Formula di Eulero o metodo dei resti: si procede per divisioni successive del numero maggiore per quello minore, sostituendo ad ogni passo il valore maggiore con il minore ed il minore col resto della divisione. Il processo termina quando il resto è 0.

Esempio: A = 34 , B = 18

passo 1:  $34 \% 18 = 16$

passo 2:  $18 \% 16 = 2$

passo 3:  $16 \% 2 = 0 \leftarrow \text{stop!}$

Risultato:  $\text{MCD} = 2$

### **Esercizio 2.**

*Competenze: IO formattato, manipolazioni di numeri;*

*Categoria: problemi numerici (Dal problema al programma: 3.1)*

Scrivere un programma in linguaggio C che visualizzi i primi N numeri della serie di Fibonacci, con N acquisito in input da tastiera.

Suggerimento: ecco i primi numeri appartenenti alla serie 0 1 1 2 3 5 8 ... In modo formale la serie si costruisce considerando la seguente relazione:  $X_i = X_{i-1} + X_{i-2}$ , con  $X_0 = 0$  e  $X_1 = 1$ ;

Approfondimento: si modifichi la serie come segue:

$X_i = X_{i-1} * X_{i-2}$ , con  $X_0 = 1$  e  $X_1 = 2$ ;

Si determini sperimentalmente (osservando i risultati ottenuti) quanti sono gli elementi di questa serie rappresentabili con variabili di tipo intero (`int`) e di tipo intero senza segno (`unsigned int`)

### Esercizio 3.

*Competenze: IO su file, manipolazioni di caratteri;*

*Categoria: problemi di codifica applicati a testi/caratteri (3.2.2)*

Un file (ad esempio: `sorgente.txt`) contiene un testo composto da un numero indefinito di righe. Notare che il testo NON contiene il carattere '\$'.

Lo scopo del programma è di ricodificare il testo sostituendo una sequenza di caratteri ripetuti (da un minimo di 2 ad un massimo di 9 caratteri ripetuti) con la seguente terna di caratteri:

`<carattere ripetuto>$<numero di ripetizioni>`

ATTENZIONE: il numero di ripetizioni non deve considerare la prima occorrenza del carattere: per esempio, AA contiene 1 ripetizione, BBB contiene 2 ripetizioni, etc.

Nel caso in cui un carattere sia ripetuto più di 9 volte, le ripetizioni devono essere spezzate in più intervalli.

Ad esempio:

- "AAAAAAAAAAAAA" deve essere codificato come "A\$9A\$2"
- "il numero 1000000 e' grande" deve essere codificato come "il numero 10\$5 e' grande"
- "ci sono 15 = ripetuti: ===== e 4 punti...." deve essere codificato come "ci sono 15 = ripetuti: =\$9=\$4 e 4 punti.\$3"

Il risultato della ricodifica sia salvato su un secondo file (ad esempio: `compresso.txt`).

Esempio:

Il contenuto del file `sorgente.txt` è:

Partenza	Destinazione	Costo
Parigi	New York	1000
Roma	Londra	700
Sidney	Los Angeles	2222

Il file di uscita `compresso.txt` conterrà:

Partenza	\$5	Destinazione	\$3	Costo
Parigi	\$9	New York	\$4	10\$2
\$4	Roma	\$9	Londra	\$5700
\$2	Sidney	\$6	Los Angeles	\$42\$3

Si scrivano due funzioni, in grado di effettuare, rispettivamente, la compressione (codifica dal file originale a quello compresso) e la decompressione (codifica dal file compresso a quello originale). I prototipi delle funzioni siano

```
int comprimi(FILE *fin, FILE *fout);  
int decomprimi(FILE *fin, FILE *fout);
```

In caso di errore le funzioni ritornano come risultato 0, diversamente ritornano il numero di caratteri scritti nel file in uscita.

Il programma main deve:

- permettere all'utente di selezionare quale operazione effettuare: compressione o decompressione (tramite un opportuno input da tastiera)
- aprire i file di input e di output
- in base alla scelta dell'utente, richiamare la funzione di compressione o decompressione sui file aperti.

Nota: Si consiglia di usare un terzo file per il testo decompresso (ad esempio: decompresso.txt) evitando così di sovrascrivere, perdendolo, il file originale sorgente.txt

#### **Esercizio 4.**

*Competenze: IO su file, manipolazioni di caratteri;*

*Categoria: problemi di codifica applicati a testi/caratteri (3.2.2)*

Un file contiene un testo composto da un numero non noto di caratteri.

Lo scopo del programma è di ricodificare il testo, generando un secondo file, nel quale i caratteri sono stati ricodificati secondo le regole seguenti:

I caratteri numerici sono ricodificati nel seguente modo:

- I caratteri numerici ('0'...'9') sono ricodificati nel carattere numerico posto k posizioni più avanti, con k che parte da 0 e viene incrementato di 1 ogni volta che un carattere numerico viene ricodificato (ATTENZIONE: gli incrementi sono effettuati MODULO 10, cioè, arrivati a 9 si riparte da 0).

Ad esempio, se il file inizia con la riga: "Il numero 248 e' pari":

- '2' (k parte da 0) deve essere codificato come '2' + 0 = '2' (k diventa 1)
- '4' deve essere codificato come '4' + 1 = '5' (k diventa 2)
- '8' deve essere codificato come '8' + 2 = '0' (dopo aver superato '9' si riparte da '0')

I caratteri alfabetici sono ricodificati nel seguente modo:

- Se un carattere alfabetico è preceduto da un carattere non alfabetico, resta inalterato
- Se è preceduto da un carattere alfabetico (sia c0 il carattere precedente), il suo codice ASCII deve essere incrementato di h posizioni nell'insieme dei caratteri alfabetici (con h=c0-'A' se c0 è maiuscolo, h=c0-'a' se c0 è minuscolo). L'incremento di h è MODULO

26, cioè arrivati alla 'z' o 'Z' (a seconda che il carattere ricodificato sia maiuscolo o minuscolo) si riparte da 'a' o 'A'.

Il risultato della ricodifica sia salvato su un secondo file (I nomi dei file siano opportunamente acquisiti da tastiera).

Esempio:

Se il contenuto del file è:

```
Apelle figlio di Apollo
fece una palla di pelle di pollo
tutti i pesci vennero a galla
per vedere la palla di pelle di pollo
fatta da Apelle figlio di Apollo.
```

Il file di uscita codificato conterrà:

```
Aptept fntema dl Apdozn
fjlp uhh ppall dl ptept dl pdozn
tngzh i ptlnv vzmzdui a ggrcc
ptk vzcgb ll ppall dl ptept dl pdozn
ffyrr dd Aptept fntema dl Apdozn.
```

Si scrivano due funzioni, in grado di effettuare, rispettivamente, la codifica (dal file di origine al file codificato) e la decodifica (dal file codificato al file di origine).

I prototipi delle funzioni siano:

```
int codifica(FILE *fin, FILE *fout);
int decodifica(FILE *fin, FILE *fout);
```

In caso di errore le funzioni ritornano come risultato 0, diversamente ritornano il numero di caratteri scritti nel file in uscita.

Il programma main deve:

- permettere all'utente di selezionare quale operazione effettuare: codifica o decodifica (tramite un opportuno input da tastiera)
- acquisire da tastiera i nomi dei file di input e output
- aprire i file di input e di output
- in base alla scelta dell'utente, richiamare la funzione di codifica o decodifica sui file aperti.