# TECNICHE DI PROGRAMMAZIONE

Esercitazione di Laboratorio 1

---

Obiettivi

- Risolvere problemi gestendo input-output

Contenuti tecnici

- Definizione di variabili intere, float, char e loro utilizzo
- Uso di *(f)scanf* e *(f)printf*, *(f)gets* e *(f)puts*, *(f)getc/getchar* e *(f)putc*
- Uso della direttiva *#define*
- Uso degli specificatore di formato (*%d*, *%f*, *%c*, *%s*)
- Lettura di dati da file
- Uso dell'operatore cast

***ATTENZIONE**: il laboratorio richiede la conoscenza del materiale presentato nelle lezioni della prima settimana (Capitolo 2 – Il C in breve), e del documento "C_IO_summary" o altro materiale equivalente. L'utilizzo di CLion, pur se consigliato, NON è un requisito, altri ambienti (IDE) adatti a sviluppare programmi C possono essere ampiamente adeguati.*

---

Da risolvere preferibilmente durante il laboratorio

Esercizio 1. Utilizzando l'ambiente di sviluppo creare un nuovo progetto e scrivere il seguente programma in linguaggio C. Verificare, inoltre, che non siano presenti errori in fase di compilazione.

```
#include <stdio.h>
int main(void)
{
    int x, y;
    float z;

    printf("Insert an integer number:");
    scanf("%d", &x);
    y = 3;
    z = (float)(x)/y;

    printf("%d/%d=%.3f\n", x, y, z);
    return 0;
}
```

Dopo aver compilato il codice, eseguirlo ed esercitarsi con l'esecuzione del debug osservando il valore delle variabili *x*, *y* e *z*, provare con diversi valori: *0*, *9*, *15*, *20*.

Esercizio 2. Utilizzando l'ambiente di sviluppo creare un nuovo progetto e creare il seguente file di testo all'interno della cartella del progetto.

Guide.txt

*"Quarantadue!" urlo' Loonquawl. "Questo e' tutto cio' che sai dire dopo un lavoro di sette milioni e mezzo di anni?"*
*"Ho controllato molto approfonditamente," disse il computer, "e questa e' sicuramente la risposta. Ad essere sinceri, penso che il problema sia che voi non abbiate mai saputo veramente qual e' la domanda."*

Scrivere il seguente programma in linguaggio C e verificare che non siano presenti errori in fase di compilazione.

```
#include <stdio.h>

int main() {
    FILE *fp_read, *fp_write;
    char file_char, choice;

    if ((fp_read = fopen("../Guide.txt", "r")) == NULL) {
        printf("Error opening file\n");
        return 1;
    }
    if ((fp_write = fopen("../Output.txt", "w")) == NULL) {
        printf("Error opening file\n");
        return 2;
    }

    printf("Print on console (C) or on file (F):");
    choice = getchar();

    while (!feof(fp_read)) {
        file_char = fgetc(fp_read);
        if (!feof(fp_read)){
            switch (choice) {
                case 'C':
                    printf("\nChar   printed   on   the   console:   %c",
file_char);
                    break;
                case 'F':
                    fputc(file_char, fp_write);
                    printf("\nChar saved on file: ");
                    putchar(file_char);
                    break;
                default:
                    printf("Wrong choice\n");
                    return 3;
            }
        }
    }

    fclose(fp_read);
    fclose(fp_write);

    return 0;
}
```

Dopo averlo compilato, eseguire il codice ed esercitarsi testando i vari case.
Approfondimento: Cosa succede nella stampa su file se non viene inserita la riga *if (!feof(fp_read))*? Perché si verifica?

Esercizio 3. Utilizzando l'ambiente di sviluppo creare un nuovo progetto e creare il seguente file di testo all'interno della cartella del progetto.

Bronte.txt

*Ho sognato nella mia vita,*
*sogni che son rimasti sempre con me,*
*e che hanno cambiato le mie idee;*
*son passati attraverso il tempo e attraverso di me,*
*come il vino attraverso l'acqua,*
*ed hanno alterato il colore della mia mente.*

Scrivere il seguente programma in linguaggio C e verificare che non siano presenti errori in fase di compilazione.

```
#include <stdio.h>

int main() {
    FILE *fp_read, *fp_write_odd, *fp_write_even;
    char file_string[100], name[20];
    int counter = 0;

    if ((fp_read = fopen("../Bronte.txt", "r")) == NULL) {
        printf("Error opening file\n");
        return 1;
    }
    if ((fp_write_odd = fopen("../Output_odd.txt", "w")) == NULL) {
        printf("Error opening file\n");
        return 2;
    }

    if ((fp_write_even = fopen("../Output_even.txt", "w")) == NULL) {
        printf("Error opening file\n");
        return 3;
    }

    printf("What's your name?");
    gets(name);

    while (!feof(fp_read)) {
        counter++;
        if (counter%2==0) {
            fscanf(fp_read, "%s", file_string);
            if (!feof(fp_read)) {
                printf("%s\nI am reading:\n%s\n\n", name, file_string);
                fprintf(fp_write_even, "%s", file_string);
            }
        }
        else {
            fgets(file_string, 100, fp_read);
            if (!feof(fp_read)) {
                puts(name);
                puts("I am reading:");
                puts(file_string);
                fputs(file_string, fp_write_odd);
```

```
            }
        }
    }

    fclose(fp_read);
    fclose(fp_write_even);
    fclose(fp_write_odd);

    return 0;
}
```

Che cosa leggo quando il contatore è pari? E quando è dispari? Quale è la differenza tra *fgets* ed *fscanf*?

Esercizio 4. Si scriva un programma che calcoli l'area di un cerchio o di un quadrato in base alla scelta dell'utente. L'utente può specificare se calcolare l'area del quadrato (*Q*) tramite la lunghezza della diagonale (D) o del lato (L), mentre l'area del cerchio (*C*) specificando se tramite il diametro (D) oppure il raggio (R).
Esempio: Se l'utente inserisce *Q D10*, il programma dovrà stampare a schermo *Area Quadrato = 50.0*

In particolare:

a) Si definisca una costante P tramite #define, e gli si assegni il valore *3.14*, (nota: con la define non si mette né l'**=** né il **;**)
b) Si acquisiscano i caratteri da tastiera.
c) Si calcoli il valore dell'area in base alla scelta dell'utente.
d) Si stampi il risultato a video.

Nota: Quadrato: *Area = L*L = D*D/2*, Cerchio*: Area = pi*R*R = pi*D*D/4*)

---

Da risolvere successivamente in autonomia

Esercizio 5. Scrivere un programma "calcolatrice" che esegua le quattro operazioni aritmetiche di base (addizione, sottrazione, divisione e moltiplicazione) tra due valori *op1* e *op2*.

Realizzare un programma C che:

a) Acquisisca da tastiera utilizzando *getchar* l'operazione da eseguire ('+', '-', '*' e '/')
b) Acquisisca da tastiera utilizzando *scanf* i due operandi (float) (es. *21.0 2.0*).
c) Stampi a schermo il carattere dell'operazione seguita dal risultato (es. */ 10.50*).

Approfondimento: Cosa succede quando *op2* è uguale a 0?

Esercizio 6. Partendo dal codice dell'esercizio precedente, scrivere un programma C che legga da file una serie di operazione da eseguire e scriva su un altro file l'operazione eseguita ed il risultato dell'operazione con due cifre decimali.

*Esempio*: di seguito un possibile contenuto del file di input *Operations.txt*:

```
+ 15.225 30.51
- 42.1 10.01
* 0.62 2.4
```

```
/ 5.0 2.5
```

Ed il file di output *Results.txt*:

```
+ 45.74
- 32.09
* 1.49
/ 2.00
```