

# **Sistemi Operativi**

## **Unità 3: Programmazione in C**

### **Esercizi**

Martino Trevisan  
Università di Trieste  
Dipartimento di Ingegneria e Architettura

## Argomenti

1. Stampa di file
2. Area di un triangolo
3. Calcolo del minimo
4. Calcolo della media
5. Somma di vettori bidimensionali

## Stampa di file

Si scriva un programma che riceve il nome di un file da riga di comando e ne stampa il contenuto.

```
#include <stdio.h>
int main(int argc, char *argv[]){
    FILE * f;
    char buffer[100];

    if (argc!=2){ /* Controllo degli argomenti */
        printf("Uso: ./stampa path\n");
        return 1;
    }

    f = fopen(argv[1], "r");
    if (f == NULL){ /* Controllo sul file */
        printf("Impossibile aprire il file\n");
        return 1;
    }

    /* Stampa finchè non termina il file */
    while (fgets(buffer, 100, f)!=NULL)
        printf("%s", buffer);

    fclose(f);
    return 0;
}
```

## Area di un triangolo

Si scriva un programma che riceve base e altezza di un triangolo da riga di comando e stampa la sua area. Base e altezza sono numeri con virgola.

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    float base, altezza;

    if (argc!=3){ /* Controllo degli argomenti */
        printf("Uso: ./area base altezza\n");
        return 1;
    }

    /* Conversione */
    base = atof(argv[1]);
    altezza = atof(argv[2]);

    /* Controllo base e altezza maggiori di 0 */
    if (base<=0 || altezza <=0){
        printf("Parametri non validi. Devono essere maggiori di 0.\n");
        return 1;
    }

    printf("Area: %f\n", base*altezza/2);

    return 0;
}
```

## Calcolo del minimo

Si scriva un programma che riceve come parametro il nome di due file:

- Il primo file è di input e contiene un intero *positivo* per riga
- Il secondo file è di output e vi viene scritto il numero minimo del file di input

Si crei una riga di comando in bash che svolge lo stesso compito, ipotizzando che il file di input sia `in.txt` e quello di output `out.txt`

## Calcolo del minimo

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    FILE * fin, *fout;
    int n, min;

    if (argc!=3){ /* Controllo degli argomenti */
        printf("Uso: ./minimo filein fileout\n");
        return 1;
    }

    fin = fopen(argv[1], "r");
    if (fin == NULL){ /* Controllo sul file */
        printf("Impossibile aprire il file in input\n");
        return 1;
    }

    fout = fopen(argv[2], "w");
    if (fout == NULL){ /* Controllo sul file */
        printf("Impossibile aprire il file di output\n");
        return 2;
    }

    fscanf(fin, "%d\n", &min); /* Valore iniziale per il minimo */
    while (fscanf(fin, "%d\n", &n) != EOF ) /* Cerca il minimo */
        if (n<min)
            min = n;

    fprintf(fout, "%d\n", min);
    fclose(fin);
    fclose(fout);
    return 0;
}
```

## Calcolo del minimo

La versione in bash è molto più compatta

```
cat in.txt | sort | head -1 > out.txt
```

## Calcolo della media

Si scriva e si testì una funzione che calcola la media di un vettore

```
#include <stdio.h>

float media(int n, float * v){
    float s = 0;
    int i;
    for (i=0; i<n; i++)
        s+=v[i];
    return s/n;
}

int main(int argc, char *argv[]){
    float lista [] = {1.5, 2.5, 4};
    printf("Media: %f\n", media(3, lista) );
    return 0;
}
```



## Somma di vettori bidimensionali

Si scriva e si testi una funzione che calcola la somma di due vettori bidimensionali.  
Se ne fornisca una versione con e senza l'uso delle `struct`.

### Senza `struct`

E' necessario l'uso dei puntatori, siccome la funzione deve ritornare due valori.

```
#include <stdio.h>

void sommaV(float x1, float y1, float x2, float y2,
            float * pxres, float * pyres)
{
    *pxres = x1+x2;
    *pyres = y1+y2;
}

int main(int argc, char *argv[]){
    float punto1x = 1.1, punto1y = 2.0, punto2x = 3.6, punto2y = 2.7;
    float puntoSx, puntoSy;

    sommaV(punto1x, punto1y, punto2x, punto2y, &puntoSx, &puntoSy);
    printf("La somma vettoriale e': (%f, %f)\n", puntoSx, puntoSy);

    return 0;
}
```

# Somma di vettori bidimensionali

Con `struct`

Utilizziamo una `typedef` per evitare di ripetere molte volte la keyword `struct`.

```
#include <stdio.h>

typedef struct {
    float x;
    float y;
} punto;

punto sommaP(punto p1, punto p2){
    punto risultato;
    risultato.x = p1.x + p2.x;
    risultato.y = p1.y + p2.y;
    return risultato;
}

int main(int argc, char *argv[]){
    punto p1 = {1.4, 4.2};
    punto p2 = {3.2, 5.9};

    punto s = sommaP(p1, p2);
    printf("La somma vettoriale e': (%f, %f)\n", s.x, s.y);

    return 0;
}
```