ESEMPIO 4

È dato un file di testo **`elenco.txt`** le cui righe rappresentano *ciascuna i dati di una persona*, secondo il seguente formato:

- **cognome** (esattamente 10 caratteri)
- **nome** (esattamente 10 caratteri)
- **sesso** (esattamente un carattere)
- **anno di nascita**

I primi due possono contenere spazi al loro interno

NB: non sono previsti spazi espliciti di separazione

ESEMPIO 4

Cosa cambia rispetto a prima?

- sappiamo esattamente dove iniziano e dove finiscono i singoli campi
- non possiamo sfruttare gli spazi per separare cognome e nome

Un possibile file **`elenco.txt`**:

```
Rossi      Mario      M1947
Ferretti   Paola      F1982
De Paoli   Gian Marco M1988
Bolognesi  Anna Rita  F1976
...
```

I vari campi possono essere "attaccati": tanto, sappiamo a priori dove inizia l'uno e finisce l'altro

ESEMPIO 4

Come fare le letture?

- **non possiamo usare `fscanf(f,"%s",…)`**
  - **si fermerebbe al primo spazio**
  - **potrebbe leggere più caratteri del necessario (si pensi a `Gian MarcoM1988`)**
- **però possiamo usare `fscanf` nell'altra modalità, *specificando quanti caratteri leggere*.** Ad esempio, per leggerne dieci:

```
fscanf(f,"%10c",…)
```

Così legge esattamente 10 caratteri, spazi inclusi

ESEMPIO 4

Come fare le letture?

- **non possiamo usare `fscanf(f,"%s",…)`**
  - **[illegible]**
  - **[illegible] cessario**
- **per[illegible] altra modalità, *specificando [illegible]ti caratteri leggere*.** Ad esempio, per leggerne dieci:

  `fscanf(f,"%10c",…)`

  Così legge esattamente 10 caratteri, spazi inclusi

ATTENZIONE: viene riempito un array di caratteri, *senza inserire alcun terminatore*
Occorre aggiungerlo a parte

ESEMPIO 4: PROGRAMMA COMPLETO

```
#define DIM 30
#include <stdio.h>
#include <stdlib.h>

typedef struct {
 char cognome[11], nome[11], sesso; int anno;
} persona;

main() {
 persona v[DIM]; int k=0; FILE *f;
 if ((f=fopen("elenco.txt", "r"))==
  perror("Il file non esiste!");
 while(fscanf(f,"%10c%10c%c%d\n", v[k].cognome,
    v[k].nome, &v[k].sesso, &v[k].anno ) != EOF){
     v[k].cognome[10]=v[k].nome[10]='\0'; k++;}
}
```

Legge esattamente 10 caratteri (spazi inclusi)

Legge 1 carattere e un intero (ricordare &)

Ricordare il terminatore!