

# Esercitazione 7

## Tipi Strutturati e Puntatori

Alberto Marchesi  
Informatica A – Ingegneria Matematica (M—Z)

21 Ottobre 2020

Gli esercizi visti a lezione sono segnalati con (\*).

### Esercizi sui Tipi Strutturati

**Esercizio 7.1.** (\*) Si consideri le dichiarazioni in Figura 1 che definiscono i tipi di dati relativi agli impiegati di una azienda, agli uffici occupati dagli impiegati, e all’edificio che li ospita (20 piani ognuno con 40 uffici).

```
typedef struct {  
    char nome[20], cognome[20];  
    int stipendio;  
    int cat;  
} Impiegato;  
  
typedef struct {  
    int superficie;  
    char esp[20];  
    Impiegato occupante;  
} Ufficio;  
  
int main () {  
  
    Ufficio torre[20][40];  
  
}
```

Figura 1: Dichiarazione tipi Esercizio 7.1.

Scrivere frammenti di codice che svolgono le seguenti operazioni:

1. stampare cognome, stipendio e categoria degli impiegati che occupano uffici orientati a sud oppure a sud-est con una superficie compresa tra 20 e 30 metri quadrati;
2. stampare i numeri dei piani che non hanno neanche un ufficio esposto a nord;

3. stampare in che piano e in che ufficio si trova *Giacomo Boracchi*;
4. copiare in un array tutti gli uffici occupati da impiegati di categoria 5;
5. copiare in un array tutti gli impiegati di categoria 5.

## Esercizi sui Puntatori

**Esercizio 7.2.** (\*) Dire cosa stampa il seguente codice.

```
#include <stdio.h>
#define LEN 5

int main() {
    char a[LEN];
    int i;
    char *p, *q;
    for (i = 0; i < LEN; i++) {
        a[i] = 'A';
    }
    p = a;
    *p += 15;
    q = p + 3;
    *q += 8;
    *(q - 1) = 76;
    p = p + 1;
    *p = (*(p - 1) - 1);
    a[LEN - 1] = '\0';
    printf("%s\n", a);
}
```

Figura 2: Codice Esercizio 5.1.

**Esercizio 7.3.** Scrivere un programma che legge una matrice di interi di dimensione  $N \times N$ , un vettore di interi di lunghezza  $N$ , e un intero  $K$ , e stampa l'indice della riga della matrice che ha esattamente  $K$  valori in comune con il vettore, se esiste,  $-1$  altrimenti. Si supponga che sia il vettore che la matrice non contengano elementi duplicati.

**Nota.** Accedere agli elementi della matrice e del vettore usando solo i relativi puntatori (non usare le operazioni di indicizzazione, come, ad esempio, `vett[i]`).

**Esercizio 7.4.** (\*) Dichiarare un tipo di dato `ElencoPunti` composto da: un vettore di punti di lunghezza massima 10, ed il numero di punti effettivamente memorizzati nel vettore.

Il tipo `Punto` é caratterizzato da un valore di  $x$  (ascissa) e di  $y$  (ordinata).

Scrivere un frammento di codice che trova il punto di ascissa massima e quello di ordinata massima, poi, se sono lo stesso punto lo stampa, altrimenti stampa la somma dei due punti.

**Nota.** Usare due puntatori per salvare i riferimenti ai due punti di interesse (evitare di salvare i due punti in ulteriori variabili).