

# Esercitazione 7

## Programmazione Avanzata ed Elementi di Ingegneria del Software

### Esercizio 1:

Scrivere una funzione che riordini un vettore di numeri interi utilizzando un algoritmo diverso fra merge, selection, insertion e heap sort a seconda della scelta dell'utente.

### Esercizio 2:

Considerato un oggetto:

```
struct Item {  
    int x;  
    char *name;  
}
```

scrivere un programma che stampi la media matematica dei caratteri del campo name ripetuta  $x$  volte.

Successivamente, considerare un secondo oggetto:

```
struct Item {  
    int x;  
    const int y = 10;  
    char *name;  
}
```

e creare un secondo programma che stampi semplicemente il contenuto di name solo se il numero di caratteri è uguale a  $x + y$ .

Il file main.c e tutti i file .h dei due programmi devono essere identici.

### Esercizio 3:

Data la funzione riportata in basso e ipotizzando che ogni azione abbia costo unitario, indicarne la complessità computazionale per  $n = 10$  e stimarne l'ordine. Cosa c'è dentro  $v3$ ?

```
unsigned long unknown (int n) {  
    unsigned long v0 = n;  
    unsigned long v1 = n / n;  
    unsigned long v2 = (2*n) / n - 1;  
    unsigned long v3 = 1;  
    unsigned long x, y, z;  
    for (x = v0; x > 0; x--) v1 *= x;  
    for (y = v1; y > 0; y--) v2 *= y;  
    for (z = v2; z > 0; z--) v3 *= z;  
    return v3;  
}
```