



# Programmazione Avanzata

Durata: 4h  
Data: 29 agosto 2024  
Punteggio Massimo: 26 PUNTI.

Chiama i tuoi progetti con MATRICOLA\_ESX con X il numero dell'esercizio. Per esempio, 2574188\_ES1 conterrà la soluzione dell'esercizio 1 dello studente con matricola 2574188.

**Consegna:** crea uno zip unico e carica il file su Moodle [cartella ``2024-08-29``].

## 1. Funzione in C (4 punti)

Scrivi una funzione in C che dato in ingresso un array di int restituisce il numero di elementi multipli di 3 contenuti nell'array. Scrivi tre versioni: una non ricorsiva, una ricorsiva senza tail recursion e una ricorsiva con tail recursion. Specifica esattamente i parametri che passi alla procedura, il tipo di passaggio utilizzato e il loro significato.

Scrivi anche un main di esempio in cui chiami le funzioni con un array di tua scelta con 10 elementi. L'array deve essere dichiarato come variabile nel main.

Non usare variabili globali.

## 2. Record di Attivazione (4 punti)

Si consideri la seguente funzione:

```
int sum_n(int a, int ex) {  
    if(ex==0) return 0;  
    return a+(sum_n(a,(ex-1)));  
};
```

Si mostrino i record di attivazione nello stack quando chiamata con a=5, ex=3: si crei un file .txt composto come segue (si aggiungano righe al file se necessario)

| Label | Indirizzo | Contenuto | Note |
|-------|-----------|-----------|------|
|       |           |           |      |
|       |           |           |      |
|       |           |           |      |



### 3. Tipi opachi (3 punti)

Definisci il tipo opaco DescSet che rappresenta un insieme (ordinato in ordine decrescente) di interi.

Nota: un insieme non contiene elementi ripetuti. Definisci l'operazione di unione. La dimensione del set deve aumentare se necessario. Implementa il set utilizzando un array. Definisci eventuali funzioni aggiuntive se necessario.

**costruttore:** crea un set ordinato e inserisce l'insieme di interi contenuti in un array passato come argomento (facendone una copia).

**union:** prende due set e restituisce un set contenente l'unione (ordinata) degli elementi.

**print:** stampa il set di interi.

**cancella:** distrugge il set e libera la memoria

Fai un esempio in cui:

- costruisci il set con gli elementi {1,3,2} e lo stampi
- Calcola l'unione dei set contenenti gli elementi {1,7,5,7} e {8,3,3,1}
- cancelli tutti i set.

### 4. Inizializer list in C++ (1.5 punti)

Definire una classe Zaino con un attributo intero che rappresenta il peso dello zaino. Definire una classe Alpinista che ha nome, cognome, e zaino come attributi. Sviluppare tre implementazioni diverse della classe Alpinista in cui l'attributo Zaino è rispettivamente (a) puntatore a una porzione di memoria nello stack, (b) puntatore a una porzione di memoria nell'Heap come puntatore, (c) reference a una porzione di memoria nell'Heap. Per ciascuno di questi tre casi discutere se (a) è possibile utilizzare l'Initializer list e (b) è possibile inizializzarlo all'interno del costruttore.

### 5. Smart pointers in C++ (1.5 punti)

Implementa la struttura dati "DescSet" dell'esercizio 3 in java, che però sia generica in modo che si possa memorizzare ogni tipo che estenda Object. Come sottostante NON usare un array list o altre collezioni, ma utilizza un array puro array[]. Implementa i metodi dell'esercizio 2. Crea un main in cui illustri l'utilizzo dell' DescSet creando un set di Integer e un set di caratteri.

### 6. Java Generics (4 punti)

Implementa la struttura dati "DescSet" dell'esercizio 3 in java, che però sia generica in modo che si possa memorizzare ogni tipo che estenda Object. Come sottostante NON usare un array list o altre collezioni, ma utilizza un array puro array[]. Implementa i metodi dell'esercizio 2. Crea un main in cui illustri l'utilizzo dell' DescSet creando un set di Integer e un set di caratteri.



## 7. Visitor pattern (4 punti)

Si vuole gestire il menu di un ristorante. Il ristorante offre varie pietanze. Ogni pietanza ha un quantitativo di calorie. Considerare due pietanze di esempio: le carote (carrot) e l'agnello (lamb). A differenza delle carote, l'agnello ha un peso ed è possibile selezionare la cottura mediante un opportuno metodo: l'agnello offre il metodo configuraCottura(String cotturaSelezionata). Le carote offrono un metodo configuraPrezzemolo che prende un booleano e permette di inserire o meno il prezzemolo nelle carote.

- (a) Si vuole utilizzare il pattern Visitor per
  - 1) stampare il menu in varie lingue (considerare per esempio l'inglese e l'italiano) e
  - 2) stampare le calorie dei vari piatti
- (b) Deve essere possibile ordinare le pietanze in ordine crescente in base alle loro calorie. Utilizzare l'interfaccia Comparable e mostrare un esempio nel quale viene creata una lista di pietanze e vengono ordinate in base alle calorie.
- (c) Creare una Lista di elementi di tipo Agnello contenente Agnelli di peso differente. Discuti se è possibile assegnare questa lista ad un Lista di elementi di tipo Pietanza e presenta le relative motivazioni.

## 8. Programmazione funzionale (4 punti)

- (a) Definire in Scala la funzione raddoppia che dato un numero intero lo moltiplica per 2, e la funzione sum che dati due numeri interi li somma.
- (b) Definire la funzione magic che data una funzione  $f: \text{Integer} \rightarrow \text{Integer}$  e una funzione  $f: \text{Integer} \times \text{Integer} \rightarrow \text{Integer}$  restituisce la funzione  $g(f(x), y)$ . Dati due numeri interi, utilizzare la funzione magic in combinazione con le funzioni definite in precedenza per sommare 3 con il doppio di 4.
- (c) Definire una list di interi contenente gli interi 1,2,3. Moltiplicare ogni elemento della lista per due, utilizzando una funzione anonima
- (d) Utilizzare map-reduce per sommare la lista dei valori ottenuti al punto (c).