UNIVERSITÀ DEGLI STUDI DI BERGAMO
Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

# Programmazione Avanzata

Durata: 4h
Data: 25 gennaio 2024

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.

Per la consegna crea uno zip unico caricare il file su Moodle nella cartella `` Programmazione Avanzata 2024-01-25''.

Punteggio Massimo: 26 PUNTI.

## 1. Funzione in C (4 punti)

Scrivi una funzione in C che dato in ingresso un array di int restituisce il numero di elementi pari 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. Descrivi anche le assunzioni che fai (ad esempio zero terminated o cose simili).

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 searchMinRec(int array[], int n){
        if (n == 1)
                return array[0];
        int min = searchMinRec (array +1, n-1);
        if (min < array [0])
                return min;
        else
                return array[0];
}
```

Si mostrino i record di attivazione nello stack quando chiamata con array =[12,13], m=2: 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 MisurazioneTemperatura che memorizza per una stazione di misurazione l'insieme delle misurazioni di temperatura registrate nelle ultime 24 ore. Definisci l'operazione per inserire una nuova misurazione. Temperature più vecchie di 24 ore vengono sovrascritte.

Proponi un implementazione del tipo un MisurazioneTemperatura che fornisce i metodi seguenti (definisci eventuali funzioni addizionali se necessario)

**costruttore**: crea una MisurazioneTemperatura. Inizialmente il valore di ogni misurazione di temperatura registrata nelle ultime 24 ore è zero.
**insert**: inserisce una nuova misurazione.
**printMisurazioni:** stampa le misurazioni delle ultime 24 ore.
**cancella**: libera la memoria
Fai un esempio in cui:

- Costruisce un nuovo elemento di tipo MisurazioneTemperatura
- Inserisce 30 misurazioni di temperatura a piacimento
- Elimina il misuratore di temperatura

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

Definire una classe Bagaglio con un attributo intero che rappresenta il peso del bagaglio. Definire una classe Passeggero che ha nome, cognome, e bagaglio come attributi. Sviluppare tre implementazioni diverse della classe Passeggero in cui l'attributo Bagaglio è rispettivamente un (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'Inizializer list e (b) è possibile inizializzarlo all'interno del costruttore.

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

Considerare la classe Passeggero dell'esercizio 4. Crea un passeggero utilizzando un raw pointer, uno smart pointer, e uno shared smart pointer. Ricordarsi che alla distruzione del passeggero deve essere distrutto anche il relativo bagaglio. Illustra e spiega le differenze relative all'utilizzo dei vari tipi di puntatori creando un metodo main e commentando opportunamente il codice.

## 6. Java Generics (4 punti)

Implementa la struttura dati "Carrozza" 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 []. La classe carrozza deve fornire i seguenti metodi:
**costruttore**: crea una carrozza.
**insert**: inserisce un elemento nella carrozza.
**print:** stampa il contenuto della carrozza.
Si creino le classi Passeggero e Animale. Passeggero estende animale Crea un main in cui si illustri l'utilizzo della classe Carrozza creando un carrozza per passeggeri, e una carrozza per animali. È possibile inserire passeggeri all'interno della carrozza di animali? E il viceversa? È possibile assegnare alla carrozza di passeggeri una carrozza di animali? E il viceversa? Mostrare un esempio in cui viene utilizzato il Wildcard e mostrarne l'utilizzo.

## 7. Visitor pattern (4 punti)

Si vuole sviluppare una applicazione per gestire una linea ferroviaria. Ci sono tre tipologie di treni: alta velocità, interregionali, locali. Ogni treno ha una partenza e una destinazione. A differenza dei treni ad alta velocità che si fermano solo alla partenza e alla destinazione, i treni interregionali e locali hanno una serie di fermate intermedie. I treni ad alta velocità, interregionali e locali hanno un massimo di 100, 150 e 200 passeggeri. Una corsa è effettuata da un treno e ha un numero totale di passeggeri che hanno preso il treno durante quella corsa (nota il numero potrebbe essere più alto nel caso di treni interregionali e locali considerato che i passeggeri possono salire e scendere nelle fermate intermedie).

Si vuole utilizzare il pattern Visitor per (a) stampare le fermate dei vari treni: per i treni ad alta velocità il nome delle fermate deve essere preceduta dalla stringa "AV -", per i treni interregionali il nome delle

fermate deve essere preceduto dalla stringa "I -", per i treni locali il nome delle fermate deve essere preceduto dalla stringa "L -". (b) stampare il numero massimo dei passeggeri dei vari treni.

Deve essere possibile ordinare le corse in ordine crescente in base al numero di passeggeri. Utilizzare l'interfaccia Comparable e mostrare un esempio nel quale viene creata una lista di corse e vengono ordinate in base ai passeggeri.

Creare una Lista di elementi di treni interregionali. Discuti se è possibile assegnare questa lista ad un Lista di elementi di tipo Treno e discuti le relative motivazioni.

## 8. Programmazione funzionale (4 punti)

(a) Definire in Scala la funzione mod che dato un numero intero restituisce 1 se è pari e zero altrimetni.
(b) Definire la funzione mult che dato un numero lo raddoppia.
(c) Definire la funzione magic: Integer->Integer che data una funzione f:Integer -> Boolean e una funzione g: Integer -> Integer restituisce una funzione che applica la funzione g al valore di input se il risultato ritornato dalla funzione f è vero, altrimenti ritorna il valore di input.
(d) Definire la funzione "h" utilizzando la funzione "magic" passandogli come input le funzioni "mod" e "mult" definite in precedenza. Utilizzare la funzione "h" sui numeri interi 2 e 3.
(e) Definire una list di interi contenente gli interi 1,2,3. Applicare ad ogni elemento della lista la funzione "h". Invece di utilizzare la funzione "h" una funzione anonima
(f) Utilizzare map-reduce per sommare la lista dei valori ottenuti al punto (e).