

Un `GestoreApp` consiste nella lista di applicazioni in esecuzione in un certo momento su un dispositivo mobile, per esempio uno smartphone o un tablet. Ogni applicazione è identificata da un nome univoco di massimo 20 caratteri e da un numero intero positivo che rappresenta la quantità di memoria, espressa in Mbyte, richiesta dall'applicazione per poter essere eseguita.

Le operazioni che possono essere effettuate su un `GestoreApp` sono le seguenti:

- **`inizializzaGestoreApp (GA) ;`** [2 pt]
Funzione che inizializza il `GestoreApp` di un dispositivo mobile. Inizialmente, non ci sono applicazioni in esecuzione.
- **`avviaApplicazione (GA, id, m) ;`** [7 pt]
Funzione booleana che implementa l'avvio di un'applicazione di nome `id` che richiede `m` Mbytes di memoria per poter essere avviata. Se un'applicazione con lo stesso nome è già in esecuzione l'operazione fallisce, la lista delle applicazioni rimane inalterata e la funzione restituisce `false`. Altrimenti, la funzione restituisce `true`.
- **`terminaApplicazione (GA, id) ;`** [7 pt]
Funzione booleana che chiude l'applicazione di nome `id`. Se non esiste una applicazione con quel nome, l'operazione fallisce, la lista delle applicazione rimane inalterata e la funzione restituisce `false`. Altrimenti, la funzione restituisce `true`.
- **`stampa (GA) ;`** [5 pt]
Funzione che stampa a video il `GestoreApp`. L'uscita stampa la quantità totale di memoria utilizzata dalle applicazioni fra parentesi quadre. Poi stampa i nomi delle applicazioni in esecuzione separate da virgola.

Esempio:

```
[450] Whatsapp, Shazam, TripAdvisor, PlayStore
```

In questo caso il `GestoreApp` sta utilizzando 450Mbytes per eseguire le applicazioni.

Le applicazioni sono `Whatsapp`, `Shazam`, `TripAdvisor` e `PlayStore`.

Se non ci sono applicazione in esecuzione, l'uscita è la seguente:

```
[0]
```

- **`chiudiApplicazioni (GA) ;`** [4 pt]
Funzione che chiude tutte le applicazioni in esecuzione.

Mediante il linguaggio C++, implementare il tipo `GestoreApp`, definito dalle precedenti specifiche. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento.

Esempio di funzione main()

```
int main( ){

    GestoreApp GA;
    inizializzaGestoreApp(GA);
    stampa(GA);
    // stampa a video
    // [0]

    avviaApplicazione(GA, "Whatsapp", 200);    //inserisce
    stampa(GA);
    // stampa a video
    // [200] Whatsapp

    avviaApplicazione(GA, "Shazam", 40);    //inserisce
    avviaApplicazione(GA, "TripAdvisor", 60); //inserisce
    avviaApplicazione(GA, "PlayStore", 100); //inserisce

    avviaApplicazione(GA, "Whatsapp", 200);    //non inserisce (duplicato)

    stampa(GA);
    // stampa a video
    // [400] PlayStore, TripAdvisor, Shazam, Whatsapp

    terminaApplicazione(GA, "Shazam");    //elimina
    stampa(GA);
    // stampa a video
    // [360] PlayStore, TripAdvisor, Whatsapp

    chiudiApplicazioni(GA);
    stampa(GA);
    // stampa a video
    // [0]

    return 0;
}
```

DOMANDE APERTE [5 pt]

1. **[2 pt]** Rappresentare il numero intero -72 in complemento a 2 su 8 bit, se rappresentabile.
2. **[2 pt]** Scrivere una funzione ricorsiva che stampa a video una sequenza di N asterischi.
Per esempio se N vale 7, stampa a video *****.
3. **[1 pt]** Dire se il seguente codice è corretto. Motivare la risposta.

```
int a[8];
for(int i=0; i<=8; i++)
    a[i] = i;
```