

Un Gantt è un diagramma utile per rappresentare le attività da svolgere all'interno di uno stesso progetto. Il numero massimo di attività per ciascun progetto è pari a 9. La durata di ciascuno progetto è inferiore o uguale a 49 mesi. Ogni attività è dotata di una descrizione, consistente in una stringa di lunghezza massima pari a 20 caratteri.

Implementare le seguenti operazioni, che operino su di un Gantt.

--- Metodi invocati nella PRIMA PARTE di main.cpp: ---

✓ **Gantt g;**

Costruttore che inizializza l'oggetto `g`. All'inizio il Gantt è vuoto e pertanto non ci sono attività né tantomeno dipendenze tra attività. Un Gantt vuoto deve apparire nel seguente modo (la semantica sarà spiegata alla descrizione dell'operatore di uscita):

```

--+-----+
|M-      M1-      M2-      M3-      M4-      |
+-----+
|123456789012345678901234567890123456789|
--+-----+
Dip:
--+-----+

```

✓ **g.aggiungiAtt(descr, mese, durata);**

Operazione che aggiunge una nuova attività al Gantt `g`. L'attività da aggiungere è caratterizzata mediante una descrizione (il parametro `descr`), di lunghezza massima pari a 20 caratteri, il mese di inizio (parametro `mese`, un intero ≥ 1 e ≤ 49) ed una durata pari al parametro `durata` (quest'ultimo espresso in mesi).

La `aggiungiAtt` deve fallire qualora il parametro `mese` sia non valido, oppure la somma di `mese` e `durata` sia maggiore di 49, per ovvie ragioni, oppure perché sono già presenti 9 azioni.

Una volta aggiunta una nuova attività, le viene assegnato (implicitamente o esplicitamente, decida lo studente) anche un *progressivo* maggiore o uguale a 1, di valore immediatamente maggiore rispetto al massimo progressivo precedentemente. Ad esempio, la prima attività verrà identificata con il progressivo 1 (e l'etichetta "A1", in fase di stampa), la seconda attività con il progressivo 2 (e l'etichetta "A2", in fase di stampa), e così via.

✓ **cout<<g;**

Operatore che mostra a video l'oggetto `g`. L'uscita deve essere nel seguente formato:

```

--+-----+
|M-      M1-      M2-      M3-      M4-      |
+-----+
|123456789012345678901234567890123456789|
--+-----+
A1|#####
A2|  #####
A3|   #####
A4|    #####
--+-----+
Dip: 3=>1 4=>2
--+-----+

```

Composizione prima riga:
due -, un +, quarantanove -, un +

Progettazione
Implementazione
Debug
Produzione

"Progettazione" è la descrizione dell'attività A1

Un solo spazio bianco tra il carattere "|" e l'inizio della descrizione dell'attività A1

In particolare l'oggetto `Gantt` in questione ha le seguenti 4 attività:

- A1 (“Progettazione”), che inizia al primo mese (M1) e dura 5 mesi (dunque fino ad M5 compreso)
- A2 (“Implementazione”), che inizia al quarto mese (M4) e dura 6 mesi (fino ad M9)
- A3 (“Debug”), che inizia al settimo mese (M7) e dura 10 mesi (fino ad M16)
- A4 (“Produzione”), che inizia al decimo mese (M10) e dura 22 mesi (fino ad M31)

e le seguenti 2 dipendenze:

- $3 \Rightarrow 1$
- $4 \Rightarrow 2$

Il significato delle dipendenze (tra attività) è chiarito nel punto successivo. È però utile informare sin d'ora che le dipendenze debbono essere mostrate tutte sulla riga “Dip:” e nello stesso ordine con cui sono state aggiunte nell'oggetto `g`.

Si notino inoltre i seguenti fatti, sempre relativi al formato di stampa a video:

- ogni riga delle attività termina con il simbolo “|”, seguito da uno spazio bianco e poi la descrizione dell'attività
- la riga contenente le dipendenze (ossia quella che inizia con “Dip:”) non ha alcun carattere di terminazione.

✓ `g.aggiungiDip(vincolata,vincolante)` ;

Operazione che aggiunge una dipendenza tra le attività. In particolare, il vincolo che l'attività avente numero d'ordine uguale a `vincolata` non può iniziare prima che sia terminata l'attività avente numero d'ordine pari a `vincolante`.

Nell'esempio precedente erano presenti le dipendenze $3 \Rightarrow 1$ e $4 \Rightarrow 2$ (che si leggono come “3 segue 1” e “4 segue 2”), ed hanno il seguente significato:

- l'attività A3 (“Debug”) non può iniziare prima che sia terminata l'attività A1 (“Progettazione”)
- l'attività A4 (“Produzione”) non può iniziare prima che sia terminata l'attività A2 (“Implementazione”)

Si noti come la `aggiungiDip` può fallire in molteplici situazioni. In tali casi l'oggetto `g` dovrà rimanere inalterato.

Alcune di queste situazioni di incompatibilità sono le seguenti (le altre sono lasciate allo studente):

- qualora non esista una attività avente progressivo pari a `vincolata` oppure a `vincolante`
- qualora il parametro `vincolata` e `vincolante` coincidano
- qualora al momento l'attività `vincolante` termini dopo l'inizio dell'attività che si vorrebbe vincolare
- ... (altri eventuali casi in cui l'aggiunta della dipendenza non sia consistente con lo stato del `Gantt`).

--- Metodi invocati nella SECONDA PARTE di `main.cpp`: ---

✓ `g.rimuoviAtt(prog)`

Operazione che rimuove dal `Gantt` l'attività avente progressivo `prog` e tutte le dipendenze che ne risultassero impattate.

Ad esempio, qualora esista la dipendenza $4 \Rightarrow 2$, questa andrebbe rimossa del tutto nel caso venga rimossa l'attività 4 o l'attività 2, mentre andrebbe aggiornata in $3 \Rightarrow 1$ qualora venga rimossa l'attività 1, perché in caso di rimozione di una attività debbono essere aggiornati i progressivi di tutte quelle aventi progressivo superiore, sottraendovi 1.

✓ `g.anticipaAtt(prog,mesi)`

Operazione che tenta l'anticipo dell'attività avente progressivo `prog` di un numero di mesi pari a `mesi` tenendo conto delle eventuali dipendenze coinvolte. In caso non sia possibile, anticipa l'attività del massimo numero possibile di mesi.

Ad esempio, qualora il Gantt sia il seguente:

```

-----+-----
|M-      M1-      M2-      M3-      M4-      |
+-----+-----
|1234567890123456789012345678901234567890123456789|
-----+-----
A1|#####          | Progettazione
A2|      #####          | Debug
A3|      #####          | Produzione
A4|          #####          | Verifica
-----+-----
Dip: 2=>1 4=>2
-----+-----

```

La chiamata alla funzione `anticipaAtt(4,30)` produrrà il seguente risultato:

```

-----+-----
|M-      M1-      M2-      M3-      M4-      |
+-----+-----
|1234567890123456789012345678901234567890123456789|
-----+-----
A1|#####          | Progettazione
A2|      #####          | Debug
A3|      #####          | Produzione
A4|          #####          | Verifica
-----+-----
Dip: 2=>1 4=>2
-----+-----

```

Ossia l'attività A4 è stata anticipata dal mese M40 al mese M16 (invece che al mese M10), per rispettare la dipendenza 4=>2 (A2 deve iniziare dopo che è eliminata A2, con A2 che termina ad M15).

✓ ~g;

Distruttore di g, qualora si necessaria la ridefinizione.

Mediante il linguaggio C++, realizzare i tipi di dato astratto Gantt, definito dalle precedenti specifiche. Non è permesso utilizzare funzionalità della libreria STL come il tipo `string`, il tipo `vector`, il tipo `list`, ecc. **Gestire le eventuali situazioni di errore.**

USCITA CHE DEVE PRODURRE IL PROGRAMMA

--- PRIMA PARTE ---

Test del costruttore:

```
-----+
|M-      M1-      M2-      M3-      M4-      |
+-----+
|123456789012345678901234567890123456789|
-----+
-----+
```

Dip:

Test della aggiungiAtt:

```
-----+
|M-      M1-      M2-      M3-      M4-      |
+-----+
|123456789012345678901234567890123456789|
-----+
-----+
```

A1|##### | Progettazione

Dip:

```
-----+
|M-      M1-      M2-      M3-      M4-      |
+-----+
|123456789012345678901234567890123456789|
-----+
-----+
```

A1|##### | Progettazione
A2| ##### | Implementazione

Dip:

```
-----+
|M-      M1-      M2-      M3-      M4-      |
+-----+
|123456789012345678901234567890123456789|
-----+
-----+
```

A1|##### | Progettazione
A2| ##### | Implementazione
A3| ##### | Debug
A4| ##### | Produzione

Dip:

Test della aggiungiDip 3=>1 e 4=>2:

```
-----+
|M-      M1-      M2-      M3-      M4-      |
+-----+
|123456789012345678901234567890123456789|
-----+
-----+
```

A1|##### | Progettazione
A2| ##### | Implementazione
A3| ##### | Debug
A4| ##### | Produzione

Dip: 3=>1 4=>2

Altro test della aggiungiDip (queste debbono fallire entrambe):

```
-----+
|M-      M1-      M2-      M3-      M4-      |
+-----+
|1234567890123456789012345678901234567890123456789|
-----+
A1|#####          | Progettazione
A2|  #####          | Implementazione
A3|   #####          | Debug
A4|    #####          | Produzione
-----+
Dip: 3=>1 4=>2
-----+
```

--- SECONDA PARTE ---

Test della rimuoviAtt:

```
-----+
|M-      M1-      M2-      M3-      M4-      |
+-----+
|1234567890123456789012345678901234567890123456789|
-----+
A1|#####          | Progettazione
A2|  #####          | Implementazione
A3|   #####          | Debug
A4|    #####          | Produzione
A5|                       ##### | Verifica
-----+
Dip: 3=>1 4=>2 5=>3
-----+
```

```
-----+
|M-      M1-      M2-      M3-      M4-      |
+-----+
|1234567890123456789012345678901234567890123456789|
-----+
A1|#####          | Progettazione
A2|   #####          | Debug
A3|    #####          | Produzione
A4|                       ##### | Verifica
-----+
Dip: 2=>1 4=>2
-----+
```

Test della anticipaAtt:

```
-----+
|M-      M1-      M2-      M3-      M4-      |
+-----+
|1234567890123456789012345678901234567890123456789|
-----+
A1|#####          | Progettazione
A2|   #####          | Debug
A3|    #####          | Produzione
A4|      #####          | Verifica
-----+
Dip: 2=>1 4=>2
-----+
```

Regole riguardanti l'autocorrezione

Nei prossimi giorni vi verrà richiesto di effettuare l'autocorrezione. L'autocorrezione dovrete effettuarla tenendo presente che deve il vostro elaborato **dovrà** riprodurre l'uscita corretta **sia della prima che della seconda parte.**

In altre parole, i docenti non correggeranno a loro volta il vostro elaborato se:

- 1) l'uscita della prima parte alla consegna era scorretta
- 2) l'uscita di prima e seconda parte, dopo la vostra autocorrezione, non è corretta.

NB: Ovviamente le correzioni che apporterete non vi consentiranno di superare la prova grazie ad un ravvedimento "fuori tempo massimo". Servono solo a velocizzare l'operazione di correzione da parte dei docenti. Inoltre è un utile esercizio didattico, in cui lo studente può prendere coscienza dei propri errori e, in certi casi, dimostrare che con poche modifiche/intergazioni, anche l'uscita della seconda parte sarebbe stata corretta.