

Il tipo di dato **GestApp** gestisce gli appuntamenti giornalieri di uno studio dentistico. Gli appuntamenti vengono presi ininterrottamente dalle ore **h1** (per esempio, 9) alle ore **h2** (per esempio, 18), con **h2>h1**, di ora in ora (cioè, alle 9, alle 10, etc.). Per ogni ora vengono programmati due pazienti, a cui viene comunicata la stessa ora (per esempio, le 9). Di ogni paziente viene memorizzato il cognome. Le operazioni che possono essere effettuate sul tipo **GestApp** sono le seguenti:

- **GestApp (h1, h2)** **Punti: 3**
Costruttore che inizializza la classe **GestApp** per gestire gli appuntamenti dalle ore **h1** alle ore **h2**;
- **~GestApp ()** **Punti: 3**
Distruttore;
- **g.fissa (cognome)** **Punti: 4**
Funzione che fissa un appuntamento per il paziente **cognome** nella prima ora disponibile. Se l'appuntamento può essere fissato, la funzione restituisce l'ora dell'appuntamento; altrimenti restituisce -1.
- **g annulla (cognome)** **Punti: 5**
Funzione che annulla l'appuntamento del paziente **cognome** e anticipa gli appuntamenti di tutti i pazienti successivi in modo da riempire il vuoto lasciato dal paziente **cognome**. La funzione restituisce **false** se l'appuntamento non può essere annullato; altrimenti restituisce **true**;
- **cout << g** **Punti: 4**
Funzione che stampa su uscita standard l'agenda degli appuntamenti. Per ogni ora viene stampata l'ora ed i cognomi dei pazienti che hanno un appuntamento per quell'ora. Per esempio
9:00 Rossi Bianchi
10:00 Neri Gialli
- **g1=g2** **Punti: 5**
L'operatore di assegnamento per il tipo **GestApp**.
- **g1==g2** **Punti: 4**
L'operatore di uguaglianza per il tipo **GestApp**. L'operatore restituisce **false** se le agende sono differenti; altrimenti restituisce **true**;

Utilizzando il linguaggio C++, realizzare il tipo di dati astratti **GestApp**, definito dalle precedenti specifiche.

NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA:

- Effettuare il **login**
Nome: studenti
Password: studenti
- Aprire il *Dev-C++* (dal Menù *Avvio* (o *Start*) nella barra degli strumenti in fondo allo schermo, selezionare *Programmi* e quindi *Dev-C++*)
- **Prima di iniziare a svolgere l'elaborato**, selezionare la voce *Identifica studente* nel menù *Strumenti* all'interno dell'ambiente *Dev-C++* e inserire i dati richiesti
- Dal menu *File* del *Dev-C++*, aprire il progetto *esainf.dev* presente nella cartella *c:\esame\esaInf*. Il progetto contiene tre file, denominati *compito.h*, *compito.cpp* e *main.cpp*
- Scrivere la dichiarazione della classe nel file *compito.h* e la definizione delle funzioni nel file *compito.cpp*. Il file *main.cpp* contiene la funzione principale *main()* che serve a verificare le funzioni scritte nel file *compito.cpp*. Il file *main.cpp* può essere modificato. **Si tenga presente, comunque, che in sede di valutazione dell'elaborato verrà considerato esclusivamente il contenuto dei file *compito.h* e *compito.cpp***
- **Per una corretta stampa dell'elaborato**, non scrivere righe di codice di lunghezza eccessiva (mantenersi entro i margini imposti dall'ambiente *Dev-C++* cioè entro la linea verticale presente alla destra della pagina);
- **Per la Consegna:**
 - Selezionare la voce *Consegna* dal menù *Strumenti* (o *Tools*) all'interno dell'ambiente *Dev-C++* e premere il tasto INVIO fino a quando non viene chiusa la finestra che è stata attivata.
 - Aspettare al proprio posto di essere chiamati per verificare la stampa del proprio elaborato, firmarlo e **consegnarlo definitivamente**.

Condizione necessaria per la correzione dell'elaborato è che le prime tre funzioni siano state implementate correttamente.