# *Fondamenti di Programmazione B*

# Prova di programmazione (C++)

# Classe Agenda

1) *(a)* Realizzare in C++ una classe di nome Data, con tre attributi, giorno, mese, anno, di tipo intero, e con i seguenti costruttori e funzioni public:

* costruttore con tre parametri, giorno, mese, anno. N.B. controllare che *1 ≤* giorno *≤ 31*, *1 ≤* mese *≤ 12,* anno *≥ 0,* e in caso contrario sollevare un'eccezione "data non valida";
* funzione operator<(d) che restituisce true se questa data (this) è precedente alla data d;
* funzione stampa() che stampa su std output la data dell’oggetto di invocazione nel formato giorno/mese/anno.

*(b)* Realizzare una classe di nome DataConOrario, derivata da Data, con un nuovo attributo ora (di tipo intero) che rappresenta l’ora del giorno. La classe offre in particolare i seguenti costruttori e funzioni public:

* costruttore con quattro parametri, giorno, mese, anno, ora (utilizzare obbligatoriamente il costruttore con tre parametri della classe Data). N.B. controllare che *0 ≤* ora *≤ 23* e in caso contrario sollevare un'eccezione "orario non valido";
* funzione operator<(d) che restituisce true se questa DataConOrario è precedente alla DataConOrario d (utilizzare obbligatoriamente l’operatore < di Data)
* funzione stampa() che stampa su std output la data con orario dell’oggetto di invocazione nel formato giorno/mese/anno – h:ore (utilizzare obbligatoriamente la funzione stampa di Data).

*(c)* Realizzare una classe di nome Agenda che rappresenta una sequenza di oggetti di tipo DataConOrario (max. *100*). La classe offre i seguenti costruttori e metodi public:

* costruttore Agenda(): crea una sequenza vuota;
* metodo add(a): aggiunge l'oggetto a di tipo DataConOrario a questa Agenda; nel caso non ci sia spazio sufficiente solleva un'eccezione “spazio non sufficiente”;
* metodo stampa(): stampa su std output i dati di tutti gli elementi dell’oggetto Agenda con un opportuno formato di stampa.

E’ inoltre prevista una funzione che ridefinisce l‘operatore << per la classe DataConOrario.

Implementare la sequenza tramite un array di 100 elementi di tipo DataConOrario.

2) Scrivere un programma principale di prova che: crea due oggetti DataConOrario di nome A e B e li inizializza con due date con orario (giorno, mese, anno, ora) fornite dall’utente tramite std input; crea un oggetto Agenda di nome D ed inserisce A e B in D (tramite il metodo add), aggiungendo prima la data più vecchia e poi quella più recente (utilizzare l’operatore < di DataConOrario); infine stampa D su std output.

Prevedere anche la “cattura” e gestione all’interno del main delle eventuali eccezioni generate dal programma.