

# Algoritmi e Calcolatori

## A.A. 2016/2017 – Progetto di Algoritmi

### Realizzazione di un Sistema automatizzato per il supporto alla refertazione medica

#### Descrizione del problema

Un sistema automatizzato per il supporto alla refertazione si avvale di un insieme di domande a risposta chiusa, predefinite da un amministratore, per costruire dinamicamente un test da sottoporre al medico refertatore (di seguito, l'utente) e guidarlo così nella definizione dello stato di salute del paziente.

La refertazione avviene nel modo seguente:

- Il sistema dispone di una *coda* di domande da sottoporre all'utente. Questa coda è inizializzata con un opportuno set di domande.
- Il sistema propone all'utente la prima domanda della coda, corredata dall'insieme di risposte ammesse. A fronte della risposta data, il sistema verifica se la risposta richiede l'aggiunta di ulteriori domande a quelle ancora da sottoporre all'utente e già presenti in "coda".
- La refertazione termina quando la coda di domande è vuota.

Si implementi un sistema in grado di rispondere alla modalità di refertazione sopra descritta e che lavori in base alle seguenti specifiche.

#### Sistema Informativo

Il sistema ottiene tutti i dati necessari al suo funzionamento da 3 diversi file (tutti indicati a riga di comando all'avvio):

1. Il primo file contiene l'elenco delle **risposte** ammesse dal sistema. Ogni risposta è identificata all'interno dell'elenco da un identificativo (ID) univoco (due risposte non possono condividere lo stesso identificativo). Ogni risposta è definita su una riga secondo la sintassi:

**<ID> <Testo Domanda>**

Si noti che il testo della risposta può, verosimilmente, contenere spazi.

2. Il secondo file contiene l'elenco delle **domande** ammesse dal sistema. Ogni domanda è identificata all'interno dell'elenco da un identificativo (ID) univoco (due domande non possono condividere lo stesso identificativo). Ogni domanda è definita su una serie di righe consecutive:

1. la prima riga definisce l'identificativo della domanda, il suo testo ed il numero di risposte ammesse secondo la sintassi:

[Q] <ID> <# Risposte> <Testo>

2. per ognuna delle risposte ammesse, una riga dedicata definisce l'identificativo della risposta da mostrare, nonché la serie di eventuali identificativi delle domande ulteriori da sottoporre all'utente a seguito della risposta; l'identificativo della risposta è seguito da un elenco di ID di domande da sottoporre.

[A] <ID risposta> <ID domanda1> <ID domanda2> ...

L'ordine con cui appaiono nel file deve essere rispettato durante la proposizione del test.

Qualora vi siano delle domande ulteriori da sottoporre all'utente, queste andranno sottoposte sempre seguendo l'ordine definito nella sequenza.

3. Il terzo file contiene l'elenco delle domande iniziali da sottoporre all'utente, seguendo l'ordine definito nel file, non appena il test avrà inizio. Ogni domanda è definita su una riga contenente l'identificativo della domanda.

Si noti che ogni volta che si fa riferimento ad un identificativo, si intende, in generale, una informazione definibile in modo da verificarne l'univocità. Esempi possono essere: numero puro, stringa di caratteri, sequenza alfanumerica, ecc.

Al presente documento, sono allegati alcuni file di esempio per ciascuna tipologia di input.

## Esecuzione

All'avvio, il sistema deve effettuare due controlli:

1. Il primo controllo, volto a verificare la correttezza *sintattica* del sistema informativo, deve certificare che tutti e tre i file sopra citati rispettino le regole grammaticali. A titolo esemplificativo, si considerino due domande (o due risposte) che condividano lo stesso identificativo come un errore, oppure la presenza di un identificativo riferito ad una domanda (o risposta) non presente nel file di descrizione correlato.
2. Il secondo controllo, volto a verificare la correttezza *semantica* del sistema informativo, deve certificare che nessuno dei possibili percorsi del test possa generare una condizione di infattibilità del test. In questo senso, si consideri come critica una situazione in cui la coda di domande non raggiunga mai lo svuotamento completo a causa dell'accodamento di una o più domande per cui sia stata fornita una risposta in precedenza.

Il fallimento di anche uno solo dei due controlli deve portare all'interruzione dell'esecuzione e ad una opportuna messaggistica di errore per l'utente.

Una corretta verifica deve invece portare il sistema a richiedere all'utente l'inserimento di un quarto file. Questo file contiene un elenco di risposte predeterminate, volte a simulare una refertazione di test. Per ogni domanda, l'utente avrà indicato una sola tra le risposte ammesse. Ogni risposta è definita su una riga contenente l'identificativo della risposta stessa, secondo la sintassi:

**<ID Domanda> <ID Risposta>**

La sequenza di risposte non è da considerarsi indicata secondo la sequenza con cui le relative domande possono essere sottoposte ed il sistema deve procedere a verificarne la correttezza (ad esempio verificando che gli identificativi di domande e risposte esistano, che la risposta sia effettivamente tra quelle possibili per quella domanda, ecc.).

Il sistema procede quindi a svolgere la simulazione del test, come da procedura definita precedentemente, mostrando a schermo un opportuno feedback all'utente. A test completato, il sistema deve confermare l'avvenuta corretta esecuzione e deve predisporre un log rappresentativo della simulazione. Il file di log, il cui nome deve essere richiesto all'utente all'avvio, conterrà l'elenco ordinato dei testi delle domande presentate all'utente e delle risposte ottenute in un formato facilmente leggibile per l'utente. Tale formato prevede 3 righe per ogni domanda proposta, secondo il seguente formato:

**<ID Domanda> <ID Risposta>**

**<Testo Domanda> <Testo Risposta>**

**New Questions: <ID DomandaI> <ID DomandaJ> ... <ID Domanda N>**

## Richieste e Descrizione dell'Elaborato

Si richiede l'implementazione dell'intero sistema, seguendo le specifiche descritte in precedenza, secondo una delle due versioni di seguito indicate.

<b>Versione A</b>	<b>Versione B</b>
<ol style="list-style-type: none"> <li>1. Interfaccia testuale per l'avvio del sistema e l'inserimento di tutti i file di input;</li> <li>2. Implementazione del controllo sull'intero sistema informativo;</li> <li>3. Implementazione della procedura di test a partire dal file;</li> <li>4. Interfaccia testuale per la fruizione della refertazione da parte di un utente umano che includa l'opzione per il salvataggio delle risposte su file (nello stesso formato compatibile con il file di test).</li> </ol>	<ol style="list-style-type: none"> <li>1. Interfaccia testuale per l'avvio del sistema e l'inserimento di tutti i file di input;</li> <li>2. Implementazione del controllo sull'intero sistema informativo;</li> <li>3. Implementazione della procedura di test a partire dal file;</li> <li>4. Interfaccia testuale per la creazione/modifica di tutti i file di input (esclusi i file di test).</li> </ol>

I gruppi possono essere composti da massimo 3 persone. Elaborati svolti da persone singole sono caldamente sconsigliati. Per l'elaborato, i criteri di valutazione comprendono:

- Rispetto delle specifiche (es: tipo di ID definibile a seconda delle esigenze);
- Algoritmi e strutture dati utilizzati;
- Flessibilità delle soluzioni adottate (es: template class, ecc.);
- Progettazione (es: organizzazione in classi, meccanismi di ereditarietà, ecc.);
- Gestione degli errori;
- Completezza delle soluzioni adottate;
- Accuratezza nella generazione di un numero opportuno di casistiche di test (set di file di input per la verifica del programma).
- Aderenza al linguaggio di programmazione (es: uso della STL, const methods, ecc.)

Per sottomettere l'elaborato, occorre creare un archivio zip con il nome

`Prog2017-matricole-componenti_versione_<lettera_versione>.zip`

e caricarlo attraverso il portale della didattica.

- La creazione del file .zip con un nome sbagliato influirà negativamente sulla valutazione.

L'archivio zip deve contenere le seguenti cartelle:

- src: contiene tutti i file sorgenti
- input: contiene tutti i file di input utili (che devono essere diversi dagli esempi pubblicati)
- doc: contiene un breve report (max. 4 pagine pdf) su:
  - struttura del programma
  - algoritmi e strutture dati utilizzati

Verrà programmata una sessione di sottomissione/discussione dei progetti in corrispondenza degli appelli d'esame (comunicata attraverso il portale della didattica). I gruppi che intendono consegnare l'elaborato dovranno sottometterlo entro una specifica data, stabilita per ogni appello, e procedere alla prenotazione dell'orale. La prenotazione avverrà tramite procedura pubblicata sul portale della didattica. L'orale sarà finalizzato alla verifica del corretto funzionamento del programma ed alla discussione del progetto, in ogni sua parte, con ciascun componente del gruppo.

**La data di consegna massima è fissata per l'ultima sessione utile prima dell'inizio delle lezioni del corso nell'A.A. 2017/2018.** Dopo tale data si dovrà fare riferimento alle regole decise per il corso in essere. Non saranno fatte eccezioni di nessun genere.

## Metodo di valutazione

Il voto finale sarà unico per tutto il gruppo e verrà per ogni componente del gruppo. Nel caso in cui il progetto non venga ritenuto sufficiente, al gruppo sarà richiesto di

rivedere l'intero progetto e sottometterlo nuovamente in una sessione successiva. La non sufficienza del progetto genera un decremento del voto finale massimo raggiungibile: ogni rifiuto diminuisce il voto massimo raggiungibile di 2 punti (es: progetto non sufficiente una volta sola, voto massimo 28; due volte, voto massimo 26).

## Esempi – In allegato al documento (file: inputs.zip)

answers.txt – File di esempio per le risposte

questions.txt – File di esempio per le domande

starting\_sequence.txt – File di esempio per la sequenza di domande iniziale

test1.txt – File di esempio per il test