

Compito Scritto di Ingegneria del Software

11 Marzo 1998 - h. 16 aula 7.9

Esercizio n. 1

Specificare mediante una rete di Petri un sistema di controllo per un'autoradio dotata di frontalino estraibile. La radio può funzionare solo quando il frontalino è inserito. La radio viene accesa (o spenta) premendo un pulsante. Se il frontalino viene estratto mentre la radio è accesa, si accende una spia rossa che si spegne solo se viene inserito il frontalino o spinto il pulsante di accensione/spegnimento.

PUNTI 5

Esercizio n. 2

Un ristorante molto esclusivo possiede 20 tavoli, che assegna, previa prenotazione, a chi ne faccia richiesta. Il ristorante è aperto tutti i giorni dell'anno corrente. Ciascun tavolo ha una capienza variabile da uno a quattro persone.

Per poter effettuare una prenotazione è necessario fornire il giorno e il numero di persone che desiderano cenare insieme (al max. 4) e un nominativo come riferimento. La stessa persona non può comparire in prenotazioni diverse per lo stesso giorno.

La prenotazione va a buon fine solo se esiste un tavolo libero per il giorno indicato che abbia la capienza richiesta (al max. 4).

Per semplicità si ipotizzi che:

- 1) Ogni tavolo venga prenotata dando il nome di una sola delle persone che pranzeranno insieme.
- 2) Le prenotazioni possano essere effettuate solo per i giorni dell'anno corrente (numerati da 1 a 365).
- 3) Una prenotazione vada a buon fine solo se esiste un tavolo libero

Si specifichi in Z un tale sistema di gestione delle prenotazioni.

PUNTI 8

Esercizio n. 3

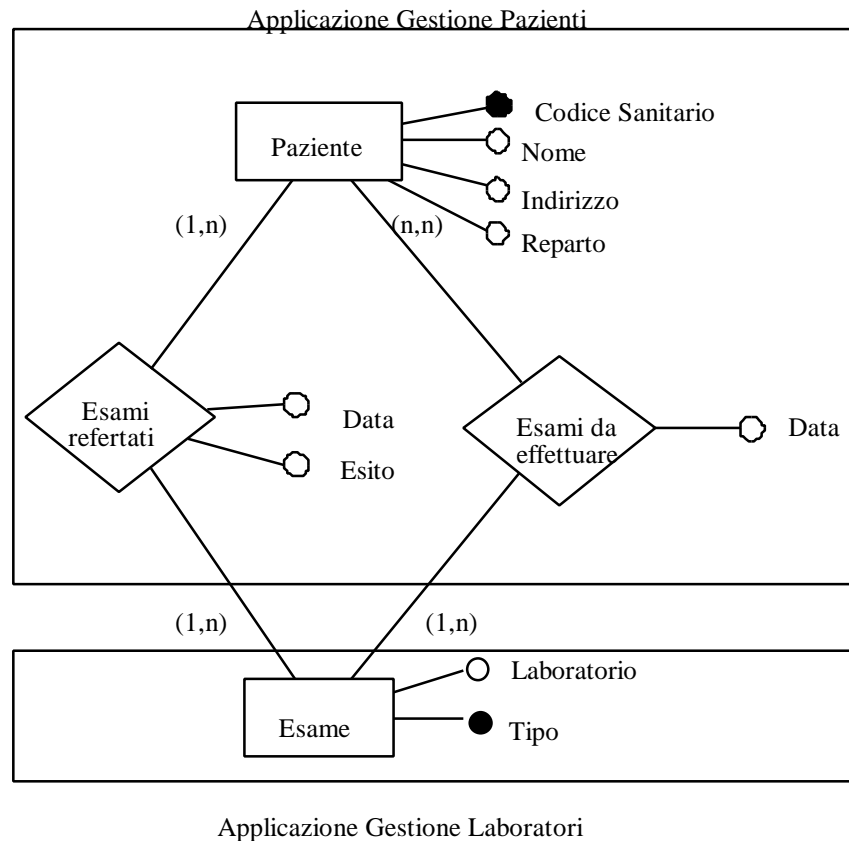
Si disegni il grafo di controllo per il programma seguente e si individui un'insieme di casi di test di dimensione minima che soddisfi il criterio di copertura delle decisioni, commentando i risultati.

```
program NATALE;  
var X,Y,Z, W : INTEGER;  
var alfa, beta: BOOLEAN;  
begin  
    alfa:= TRUE;  
    beta:= FALSE;  
    READ(X);  
    READ(Y);  
    if X < Y  
        then  
            begin  
                alfa := FALSE;  
                Z := Y-X  
            end  
        else Z := X-Y;  
    if Y > 0  
        then  
            beta := TRUE;  
    while X <= Y do  
        begin  
            Z := Z+1;  
            if beta  
                then Y:=Y-1  
                else Y:=Y+1;  
        end;  
    if not (alfa or beta )  
        then W := -Z  
        else W := Z;  
    WRITE(Z);  
    WRITE(W);  
end.
```

PUNTI 7

Esercizio n. 4

Si consideri l'applicazione Gestione Pazienti di un ospedale. L'organizzazione dei dati è rappresentata dal seguente diagramma ER:



Le linee tratteggiate indicano i confini dell'applicazione Gestione Pazienti, che si deve contare, e dell'applicazione Gestione Laboratori.

L'applicazione Gestione Pazienti deve eseguire le seguenti operazioni:

- 1 Gestione dei dati sui Pazienti
 - 1.1 Iscrizione Paziente: inserisci i dati di un nuovo paziente. I dati comprendono il numero di Codice Sanitario, il Nome, l'Indirizzo e il Reparto in cui viene ricoverato il paziente.
 - 1.2 Interroga Pazienti: richiedi i dati di un paziente dato il suo Nome o il suo numero di Codice Sanitario.
 - 1.3 Elenca Pazienti per Reparto: mostra la lista di tutti i Pazienti di un certo Reparto, stampandone il numero totale alla fine.
- 2 Gestione degli Esami
 - 2.1 Inserisci Esame da Effettuare: inserisci gli esami che un Paziente deve effettuare. L'utente deve inserire il Codice Sanitario del paziente, il Tipo di esame e la Data dell'esame di laboratorio da effettuare.
 - 2.2 Interroga Esami: richiedi l'elenco degli esami da effettuare su un Paziente dato il suo Codice Sanitario;
- 3 Gestione dei Referti
 - 3.1 Inserisci Referto: inserisce il referto di un esame di laboratorio. L'utente inserirà il Tipo dell'esame, la Data e il Referto (positivo o negativo).
 - 3.2 Interroga Esami refertati: richiede i referti degli esami effettuati da un Paziente, dato il suo numero di Codice Sanitario, e stampa il numero totale di esami refertati.

Si richiede di identificare le funzioni di tipo dati e di tipo transazione e di calcolare la complessità delle funzioni di tipo dati. Si tenga presente che Esame è considerato come ILF per l'Applicazione Gestione Laboratori.

PUNTI 10

Soluzione.

Regole di conteggio ILF	Entità e relazioni		
	Paziente	Esami da effettuare	Esami refertati
1) Il gruppo di dati o informazioni di controllo è un gruppo di dati logico, o identificabile dall'utente, che soddisfa determinati requisiti utente.	Sì	Sì	Sì
2) Il gruppo di dati è mantenuto all'interno del confine dell'applicazione.	Sì	Sì	No
3) Il gruppo di dati è modificato, o mantenuto, attraverso un processo elementare dell'applicazione.	Sì, dal processo 1.1 Iscrizione Paziente.	Sì, dal processo 2.1 Inserisci Esame da effettuare	Sì, dal processo 3.1 Inserisci Referto
4) Il gruppo di dati identificato non è stato contato come un EIF per l'applicazione.	Sì	Sì	No

Regole di conteggio EIF	Entità e relazioni
	Tipo Esame
1) Il gruppo di dati o informazioni di controllo è un gruppo di dati logico, o identificabile dall'utente, che soddisfa determinati requisiti utente.	Sì
2) Il gruppo di dati è referenziato dall'applicazione che si sta misurando ed è ad essa esterno.	Sì
3) Il gruppo di dati non è mantenuto dall'applicazione che si sta misurando.	Sì
4) Il gruppo di dati è contato come un ILF per almeno un'altra applicazione.	Sì
5) Il gruppo di dati identificato non è stato contato come un ILF per l'applicazione.	Sì

RET:

ILF o EIF	RET
Paziente	1
Esami da effettuare	1
Esami refertati	1

DET

ILF o EIF	Campo riconoscibile dall'utente e non ricorsivo? Conta 1 DET	Chiave esterna? Conta 1 DET	Conta implementazione come un DET:
Paziente			
Codice Sanitario	Sì	No	No
Nome	Sì	No	No
Indirizzo	Sì	No	No
Reparto	Sì	No	No
Totale	4		
Esami da effettuare			
Data	Sì	No	No
Codice Sanitario	No	Sì	No
Tipo	No	Sì	No
Totale	1	2	
Esami refertati			
Data	Sì	No	No
Codice Sanitario	No	Sì	No
Tipo	No	Sì	No
Totale	1	2	
Esame			
Tipo	Sì	No	No
Laboratorio	Sì	No	No
Totale	2		

Regole di conteggio EI	Processi		
	Iscrizione Paziente	Inserisci Esame da Effettuare	Inserisci Referto
1) I dati sono ricevuti dall'esterno del confine dell'applicazione.	Sì	Sì	Sì
2) I dati in un ILF sono mantenuti attraverso un processo elementare dell'applicazione.	Sì	Sì	Sì
3) Il processo è la più piccola unità di attività che è significativa per l'utente.	Sì	Sì	Sì
4) Il processo è autonomo e lascia l'applicazione che si sta misurando in uno stato di coerenza funzionale.	Sì	Sì	Sì
5) E' applicabile una delle seguenti due regole:			
i) il trattamento logico è unico rispetto ad altri EI,	Sì	Sì	Sì
ii) i dati identificati sono diversi rispetto ad altri EI.	Sì	Sì	Sì

Regole di conteggio EO	Processi	
	Elenca Pazienti per Reparto	Interroga Esami Refertati
1) Il processo invia dati all'esterno del confine dell'applicazione.	Sì	Sì
2) I dati o le informazioni di controllo sono inviati attraverso un processo elementare dell'applicazione.	Sì	Sì
3) Il processo è la più piccola unità di attività che è significativa per l'utente.	Sì	Sì
4) Il processo è autonomo e lascia l'applicazione che si sta misurando in uno stato di coerenza funzionale.	Sì	Sì
5) E' applicabile una delle seguenti due regole:		
i) il trattamento logico è unico rispetto ad altri EO,	Sì	Sì
ii) i dati identificati sono diversi rispetto ad altri EO.	Sì	Sì

Interroga Esami Refertati è un EO perché stampa un dato derivato (totale esami refertati).

Regole di conteggio EQ	Processi	
	Interroga Pazienti	Interroga Esami
1) Una richiesta di input entra nel confine dell'applicazione.	Sì	Sì
2) Un risultato di output esce dal confine dell'applicazione.	Sì	Sì
3) Sono reperiti dei dati.	Sì	Sì
4) I dati reperiti non contengono dati derivati.	Sì	Sì
5) Il processo è la più piccola unità di attività che è significativa per l'utente.	Sì	Sì
6) Il processo è autonomo e lascia l'applicazione che si sta misurando in uno stato di coerenza funzionale.	Sì	Sì
7) L'elaborazione non aggiorna alcun ILF.	Sì	Sì
8) E' applicabile una delle seguenti due regole:		
i) il trattamento logico è unico rispetto ad altri EQ,	Sì	Sì
ii) i dati identificati sono diversi rispetto ad altri EQ.	Sì	Sì

I^o Compito Intermedio di Ingegneria del Software - A.A. 97/98

23 Aprile 1998 - h. 17 aula 6.2

Esercizio 1

Specificare mediante una rete di Petri il funzionamento di sistema composto da un interruttore a pressione e una coppia di lampade. Quando entrambe le lampade sono spente, premendo il pulsante viene accesa la prima; premendo nuovamente il pulsante la prima viene spenta e la seconda accesa; premendo nuovamente il pulsante entrambe vengono accese e infine una nuova pressione del pulsante provoca lo spegnimento di entrambe.

Esercizio 2

Si dia una specifica in Z di uno sportello Bancomat avente le seguenti caratteristiche. Per poter effettuare una operazione allo sportello (prelievo o saldo) si fornisce un numero di conto corrente (intero) e un codice segreto (intero). Il sistema verifica che il codice fornito sia effettivamente quello associato al numero di conto corrente inserito.

Se la verifica va a buon fine il sistema permette di effettuare:

- un prelievo di contanti (in quantita' fissa, ad esempio Lit 100.000)
- una richiesta di saldo del conto

In particolare devono essere specificate in Z le seguenti 4 operazioni:

- a) Accesso(Numero_conto, Numero_segreto). L'operazione di Accesso andata a buon fine ha come effetto quello di bloccare lo sportello assegnandolo al codice specificato e proibendo ulteriori accessi da parte di altri utenti con codice diverso.
- b) Preleva() che restituisce il valore della cifra prelevata (Lit. 100.000) e modifica il valore del saldo. Tale operazione ha effetto solo se è stata preceduta da un'operazione di Accesso andata a buon fine (ovvero lo sportello è assegnato a un utente).
- c) Saldo() che restituisce il valore del saldo del conto corrente. Tale operazione ha effetto solo se è stata preceduta da un'operazione di Accesso
- d) Fine() che sblocca il sistema precedentemente bloccato da un'operazione di accesso.

Esercizio 3

Specificare mediante TRIO il comportamento di un canale di comunicazione aventi le seguenti caratteristiche:

"Ogni segnale entrato nel canale deve uscire al più dopo 5 secondi. Il canale non deve perdere alcun segnale e non deve generare segnali spuri."

Esercizio 4

Spiegare se e come sia possibile in C++ far sì che quando una classe M eredita da un'altra classe P i metodi di M:

- a) non possano accedere alla parte di attributi e metodi nascosti di P;
- b) possano accedere alla parte di attributi e metodi nascosti di P, ma i metodi di altre classi (escluse quelle derivate da M) non possano.

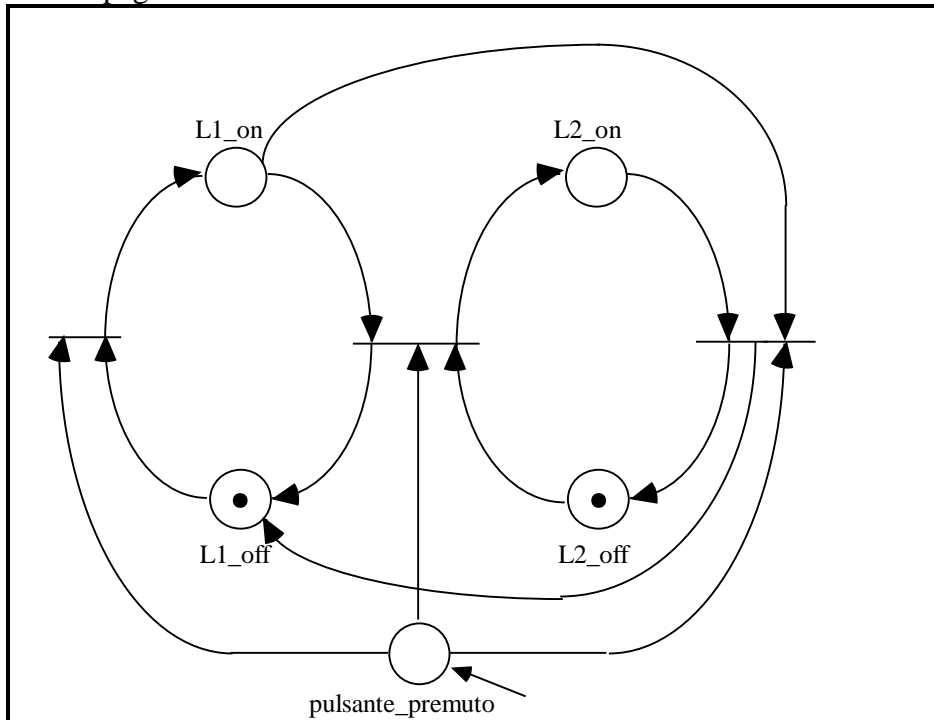
I^o Compito Intermedio di Ingegneria del Software - A.A. 97/98

23 Aprile 1998 - h. 17 aula 6.2

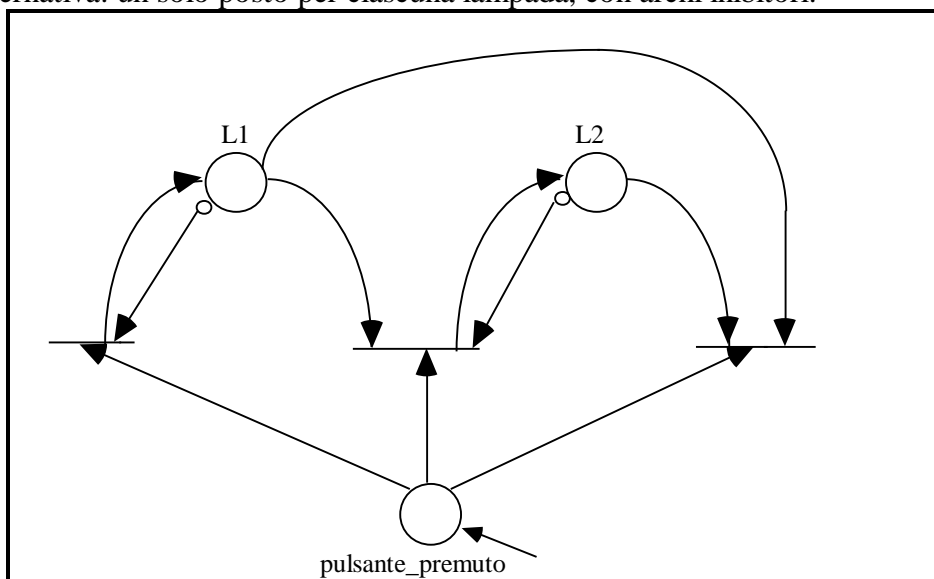
Soluzione:

Esercizio 1

Specificare mediante una rete di Petri il funzionamento di sistema composto da un interruttore a pressione e una coppia di lampade. Quando entrambe le lampade sono spente, premendo il pulsante viene accesa la prima; premendo nuovamente il pulsante la prima viene spenta e la seconda accesa; premendo nuovamente il pulsante entrambe vengono accese e infine una nuova pressione del pulsante provoca lo spegnimento di entrambe.



Soluzione alternativa: un solo posto per ciascuna lampada, con archi inibitori:



Esercizio 3

"Ogni segnale entrato nel canale deve uscire al più dopo 5 secondi. Il canale non deve perdere alcun segnale e non deve generare segnali spuri."

La seguente specifica TRIO cattura correttamente il comportamento di un tale canale di comunicazione:

$$Alw(in \wedge WithinF(out,5))$$
$$\forall m ((in(m) \rightarrow AlwF(\neg in(m)) \wedge AlwP(\neg in(m))) \wedge (out(m) \rightarrow AlwF(\neg out(m)) \wedge AlwP(\neg out(m))))$$

Esercizio 4

Spiegare se e come sia possibile in C++ far sì che quando una classe M eredita da un'altra classe P i metodi di M:

- non possano accedere alla parte di attributi e metodi nascosti di P;
- possano accedere alla parte di attributi e metodi nascosti di P, ma i metodi di altre classi (escluse quelle derivate da M) non possano.

```
class P
{public:
    int    a;        //accessibile a tutti
protected :
    int    b;        //accessibile solo alle
                    //classi derivate
private :
    int    e;        //inaccessibile
};
class M: protected P
    //indica la superclasse e la
    // visibilità di ciò che è ereditato
{public:
    int    b;
    void show()
        {Base::show();
        cout << "b:" << b << "\n";}
};
```

Scritto di Ingegneria del Software

18 Giugno 1998 - h. 10 aula 6.1 (durata 3 ore)

Esercizio 1

Specificare mediante una rete di Petri il funzionamento di un sistema di accesso a un parcheggio, costituito da un lettore di tessera magnetica, una sbarra e una cellula fotoelettrica. Quando la sbarra è chiusa, per farla alzare occorre inserire la tessera magnetica nel lettore. A questo punto la sbarra si alza e viene riabbassata quando la cellula fotoelettrica rileva il passaggio di un veicolo. Quando la fotoelettrica rileva anche il passaggio di un primo veicolo, un successivo inserimento di una tessera a sbarra alzata (da parte di un secondo veicolo) lascia alzata la sbarra. La sbarra si riabbasserà quando verrà rilevato il passaggio del secondo autoveicolo.

(Si supponga che non si possano accodare più di due auto per l'ingresso).

Inoltre il rilevamento da parte della cellula fotoelettrica del passaggio di un autoveicolo a sbarra abbassata lascia la sbarra in tale stato.

PUNTI 6

Esercizio n. 2

Un sistema di accesso a un parcheggio è costituito da un lettore di tessera magnetica, una sbarra e una cellula fotoelettrica.

Si scriva la specifica TRIO della seguente porzione di sistema:

- 1) La sbarra si alza dopo 15 sec. dall'inserimento della tessera, a meno che in tale intervallo la cellula fotoelettrica non abbia rilevato il passaggio di un veicolo;
- 2) La sbarra si riabbassa dopo un minuto.

PUNTI 3

Esercizio 3

Si dia una specifica in Z di un sistema per la registrazione di esami in forma elettronica. Ciascun docente è dotato di un tesserino magnetico con un codice identificativo e un codice segreto (entrambi interi positivi). Ciascuno studente è dotato di un tesserino magnetico contenente la sua matricola (intero positivo).

Ciascun docente può registrare solo gli esami relativi ai corsi che gli sono stati assegnati per l'anno accademico corrente. Ciascun corso è identificato da una stringa di 6 lettere.

Per poter effettuare una registrazione d'esame, la stazione d'esame deve essere disponibile (non assegnata ad alcun docente). Il docente deve fornire al sistema la propria tessera magnetica, digitare il proprio codice segreto e la stringa identificativa del corso per il quale desidera fare una registrazione.

Il sistema verifica che il codice segreto fornito sia effettivamente quello associato al numero di tessera del docente e che il docente sia abilitato a registrare esami per il corso specificato.

Se la verifica va a buon fine il sistema permette di effettuare:

- una registrazione di un voto per uno studente
- la chiusura della sessione d'esame

Altrimenti, se la verifica non ha successo si richiede al docente di inserire un altro codice segreto o un altro nome di corso.

Si specifichino in Z le seguenti operazioni:

- a) AperturaSessione(CodiceTessera,CodiceSegreto,NomeCorso).

L'operazione di AperturaSessione va a buon fine se il codice segreto fornito è quello associato al numero di tessera del docente e se il docente è abilitato a registrare esami per il corso specificato. La stazione d'esame risulta così assegnata al docente per registrare esami del corso specificato.

- b) Registra(TesseraStudente,Voto), registra il Voto dello studente con una certa TesseraStudente per il corso oggetto della sessione corrente.
- c) Chiusura(), chiude la sessione d'esame restituendo la tessera al docente e liberando la stazione d'esame bloccata da un'operazione di AperturaSessione fatta precedentemente. PUNTI 8

Esercizio n. 4

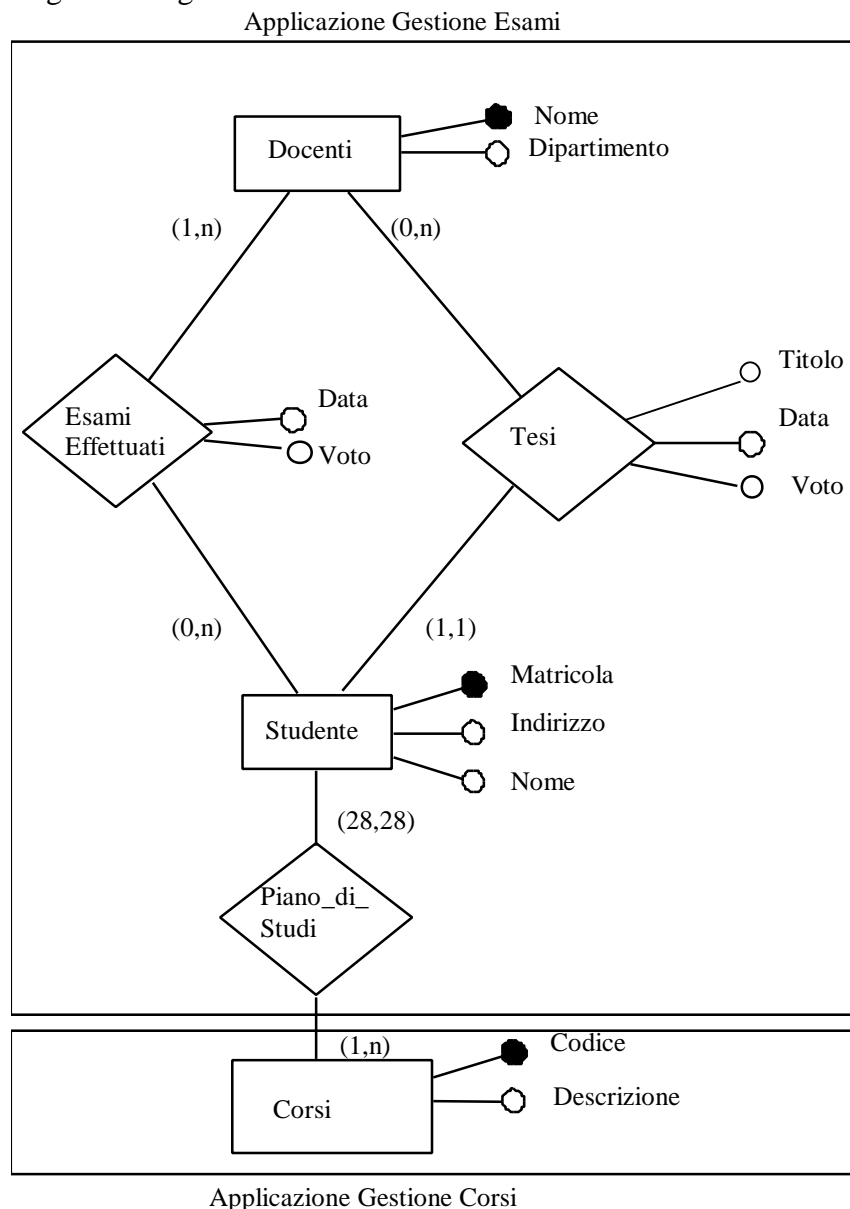
Dato il seguente programma, si determinino i cammini (minimi) da coprire selezionati dal criterio di copertura di tutti gli usi.

1	program diciotto;
2	var A,B,C: Integer;
3	X,Y: real;
4	begin
5	Read(A);
6	Read(B);
7	if (B-A*C)>0
8	then begin
9	X:=-B+sqrt(B-A*C);
10	Y:=-B-sqrt(B-A*C);
11	end
12	else X:=-Y;
13	while A-B > 0
14	do begin
15	A:= A-C;
16	X:=X-1;
17	Y:=Y-2;
18	end
19	end.

PUNTI 4

Esercizio 5

Si consideri l'applicazione Gestione Esami di una facoltà universitaria. L'organizzazione dei dati è rappresentata dal seguente diagramma ER:



Le linee tratteggiate indicano i confini dell'applicazione Gestione Esami, che si deve contare, e dell'applicazione Gestione Corsi.

L'applicazione Gestione Esami deve eseguire le seguenti operazioni:

1. Gestione dei dati sui Docenti
 - 1.1. Inserisci Docente: inserisci i dati di un nuovo docente. I dati comprendono il Nome e il Dipartimento di appartenenza.
 - 1.2. Interroga Docenti: richiedi i dati di un Docente dato il suo Nome.
 - 1.3. Elenca Docenti: mostra la lista di tutti i docenti.
2. Gestione dei dati sugli Studenti
 - 2.1. Inserisci Studente: inserisci i dati di un nuovo studente. I dati comprendono il Nome, la Matricola, l'Indirizzo e i codici dei corsi (28 in totale) inseriti nel piano di studi.
3. Gestione degli Esami e delle Tesi
 - 3.1. Inserisci Esame: inserisci i dati di un esame effettuato. I dati comprendono il Nome del docente, la Matricola dello studente, la Data e il Voto assegnato.
 - 3.2. Inserisci Tesi: inserisci i dati di una tesi sostenuta. I dati comprendono il Titolo della tesi, il Nome del docente relatore, la Matricola dello studente, la Data e il Voto assegnato.

3.3. Interroga Esami: specificato un Dipartimento come input, stampa per ogni docente del dipartimento il Nome e il numero totale di esami effettuati.

3.4. Interroga tesi: dato il Nome di un docente, stampa un elenco con Titolo, Data e Voto di tutte le tesi che hanno quel docente come relatore.

Si richiede di identificare le funzioni di tipo dati e di tipo transazione e di calcolare la complessità delle funzioni di tipo dati.

Si calcoli inoltre la complessità del processo Interroga Esami.

Si tenga presente che Corsi è considerato come ILF per l'Applicazione Gestione Corsi.

PUNTI 9

Dato il seguente programma, si determinino i cammini (minimi) da coprire selezionati dal criterio di copertura di tutti gli usi.

1	program diciotto;
2	var A,B,C: Integer;
3	X,Y: real;
4	begin
5	Read(A);
6	Read(B);
7	if (B-A*C)>0
8	then begin
9	X:=-B+sqrt(B-A*C);
10	Y:=-B-sqrt(B-A*C);
11	end
12	else X:=-Y;
13	while A-B > 0
14	do begin
15	A:= A-C;
16	X:=X-1;
17	Y:=Y-2;
18	end
19	end.

SOLUZIONE

		def	use	du(A)	du(B)	du(C)	du(X)	du(Y)
1	program diciotto;							
2	var A,B,C: Integer;							
3	X,Y: real;							
4	begin							
5	Read(A);	A		7,9, 10,13, 15				
6	Read(B);	B			7,9, 10,13			
7	if (B-A*C)>0		A,B,C					
8	then begin							
9	X:=-B+sqrt(B-A*C);	X	A,B,C				16	
10	Y:=-B-sqrt(B-A*C);	Y	A,B,C					17
11	end							
12	else X:=-Y;	X	Y				16	
13	while A-B > 0		A,B					
14	do begin							
15	A:= A-C;	A	A,C	15				
16	X:=X-1;	X	X				16	
17	Y:=Y-2;	Y	Y					17
18	end							
19	end.							

I cammini da coprire sono quindi:

- 1) 5-6-7-8-9-10-11-13-14-15 e 15-16-17-18-13-14-15 per la var. A
- 2) 6-7-8-9-10-11-13 per la var. B
- 3) 9-10-11-13-14-15-16, 12-13-14-15-16 e 16-17-18-13-14-15-16 per la var. X
10-11-13-14-15-16-17 e 17-18-13-14-15-16-17 per la var. Y

Scritto di Ingegneria del Software

14 Luglio 1998 - h. 14.30 aula E1B (durata 3 ore)

Esercizio 1

Specificare mediante una rete di Petri il funzionamento di un sistema di allarme, dotato di un pulsante di attivazione/disattivazione, un sensore acustico e una sirena. Il sistema di controllo si attiva premendo il pulsante. A sistema attivato, se il sensore acustico rileva un rumore si attiva la sirena. La sirena si spegne quando viene nuovamente premuto il pulsante. La pressione del pulsante a sirena accesa non disinserisce il sistema di controllo.

Il sistema di controllo viene disinserito, da inserito, quando viene premuto il a sirena spenta.

PUNTI 6

Esercizio n. 2

Un sistema di allarme e` costituito da un sensore acustico e da una sirena. Il sistema puo' essere inserito o disinserito.

Si scriva la specifica TRIO della seguente porzione di sistema:

- 3) La sirena si accende dopo 10 sec. dal rilevamento di un rumore da parte del sensore acustico, purché in tale intervallo non sia stato disinserito il sistema;
- 4) La sirena si spegne dopo tre minuti.

PUNTI 4

Esercizio 3

Si dia una specifica in Z di un Centro Unificato Prenotazioni (CUP) di esami di laboratorio. Ciascun utente e` dotato di un tesserino magnetico con un codice identificativo (intero positivo). Le prenotazioni avvengono presso le stazioni CUP disseminate nella citta`. Ciascuna stazione e' caratterizzata da un suo codice, anch'esso intero positivo.

Settimanalmente vengono riempite le agende dei vari laboratori che specificano, per ciascuna data e ciascuna tipologia d'esame, il numero di posti ancora disponibili.

Per poter effettuare una prenotazione d'esame, viene dato il numero della stazione CUP, inserito il codice del tesserino utente, specificata la data e il tipo di esame (stringa).

Il sistema verifica che il codice fornito sia quello associato a un utente e ci siano posti disponibili per la data e l'esame specificato. Se la verifica va a buon fine, si aggiorna l'agenda e si conferma la prenotazione. In caso contrario, si manda un messaggio di errore.

Si specifichi in Z l'operazione di prenotazione di un esame di laboratorio. PUNTI 6

Esercizio n. 4

Si individui un insieme di casi di test di dimensione minima che soddisfi il criterio di copertura delle decisioni per il programma seguente:

```
program ESAME;  
var A, B, C : INTEGER;  
var C1,C2 : BOOLEAN;  
begin  
    C1 := false;  
    C2 := true;  
    READ(A);  
    READ(B);  
    if A >= 0  
    then C1:= true  
    else A:=-A;  
    if B <= 0  
    then  
        begin  
            C2 := false;  
            B := -B  
        end;  
    C := 0;  
    while A <= B do  
        begin  
            A := A + B;  
            C := C + 1;  
        end;  
    if (C1 and C2)  
    then WRITE(C)  
    else WRITE(-C);  
end.
```

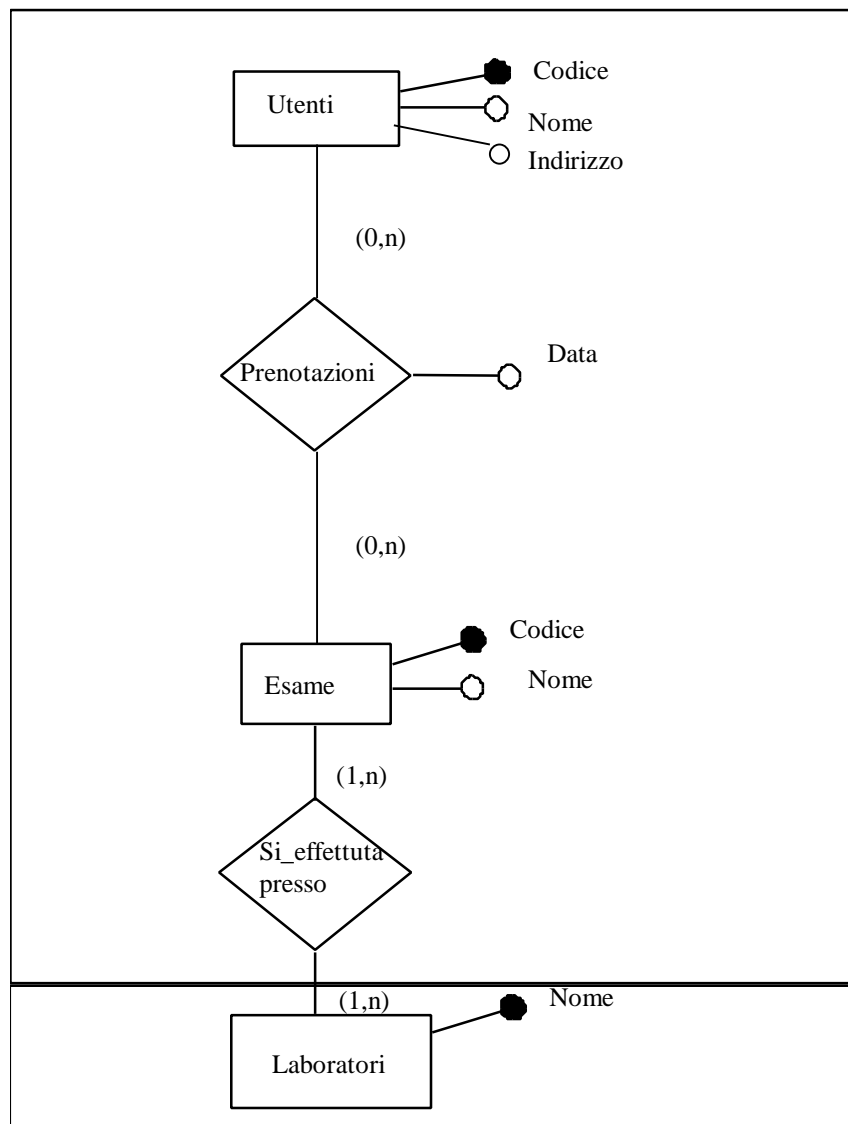
Provare ad eseguire la procedura usando i casi di test individuati e si commentino i risultati.

PUNTI 5

Esercizio 5

Si consideri l'applicazione Gestione Prenotazioni di un Centro Unificato Prenotazioni (CUP).
L'organizzazione dei dati è rappresentata dal seguente diagramma ER:

Applicazione Gestione Prenotazioni



Applicazione Gestione Laboratori

Le linee tratteggiate indicano i confini dell'applicazione Gestione Prenotazioni, che si deve contare, e dell'applicazione Gestione Laboratori.

L'applicazione Gestione Prenotazioni deve eseguire le seguenti operazioni:

4. Gestione dei dati Utenti

4.1. Inserisci Utente: inserisci i dati di un nuovo utente. I dati comprendono il Nome, il Codice CUP e l'Indirizzo.

4.2. Interroga Utenti: richiedi i dati di un Utente dato il suo Codice CUP.

4.3. Elenca Utenti: mostra la lista di tutti gli utenti.

5. Gestione dei dati sugli Esami

5.1. Inserisci Esame: inserisci i dati di un esame. I dati comprendono il Nome dell'esame, il Codice e i Nomi dei laboratori presso cui si può effettuare l'esame.

6. Gestione delle Prenotazioni

6.1. Inserisci Prenotazione: inserisci i dati di una prenotazione. I dati comprendono il Codice dell'utente, il Codice dell'esame e la Data;

6.2. Interroga Prenotazioni: specificato un Codice di esame come input, stampa per ogni utente prenotato per quel tipo di esame il Nome e la Data.

Si richiede di identificare le funzioni di tipo dati e di tipo transazione e di calcolare la complessità delle funzioni di tipo dati.

Si calcoli inoltre la complessità del processo Interroga Prenotazioni.

Si tenga presente che Laboratori è considerato come ILF per l'applicazione Gestione Laboratori.

PUNTI 9

Compito Scritto di Ingegneria del Software

23 Ottobre 1998 - h. 16 Aula 6.2

Tempo a disposizione: 3 ore

Esercizio n. 1

Modellare con una Rete di Petri un sistema di controllo degli accessi costituito da una pedana conta persone (che in base ai kg stabilisce approssimativamente il numero di persone), due porte e un allarme. La pedana si trova tra le due porte. La prima porta è solitamente aperta e la seconda chiusa.

Quando la pedana conta una persona, la prima porta si chiude e si apre la seconda porta.

Quando la pedana conta due o più persone, viene attivato l'allarme. L'allarme resta attivo finché la pedana non conta di nuovo zero persone (ovvero le persone sono scese).

PUNTI 5

Esercizio n. 2

Modellare in TRIO il funzionamento di un allarme che si attiva quando scatta un segnale e si disattiva dopo trenta secondi.

PUNTI 2

Esercizio n. 3

Un ambulatorio dentistico ha sei poltrone. Nell'ambulatorio lavorano vari odontoiatri. Ciascuno di loro può lavorare su una o due poltrone. Le poltrone sono assegnate giornalmente, previa prenotazione, a chi ne fa richiesta. Le poltrone sono disponibili dalle 8.00 sino alle 19.00 per periodi di un'ora.

Per poter effettuare una prenotazione è necessario fornire il nome dell'odontoiatria, l'ora e la durata dell'intervento (una, due o tre ore al massimo). Un odontoiatra non può comparire in più di due prenotazioni di poltrone diverse per la stessa ora (ovvero non può occupare più di due poltrone).

La prenotazione va a buon fine solo se esiste una poltrona libera per l'ora e la durata indicata.

Per semplicità si ipotizzi che:

- 1) Le prenotazioni possano essere effettuate solo per il giorno corrente;
- 2) L'ora sia individuata da un intero (compreso fra 8 e 19).

Si specifichi in Z un tale sistema di gestione delle prenotazioni delle poltrone odontoiatriche.

PUNTI 8

Esercizio n. 4

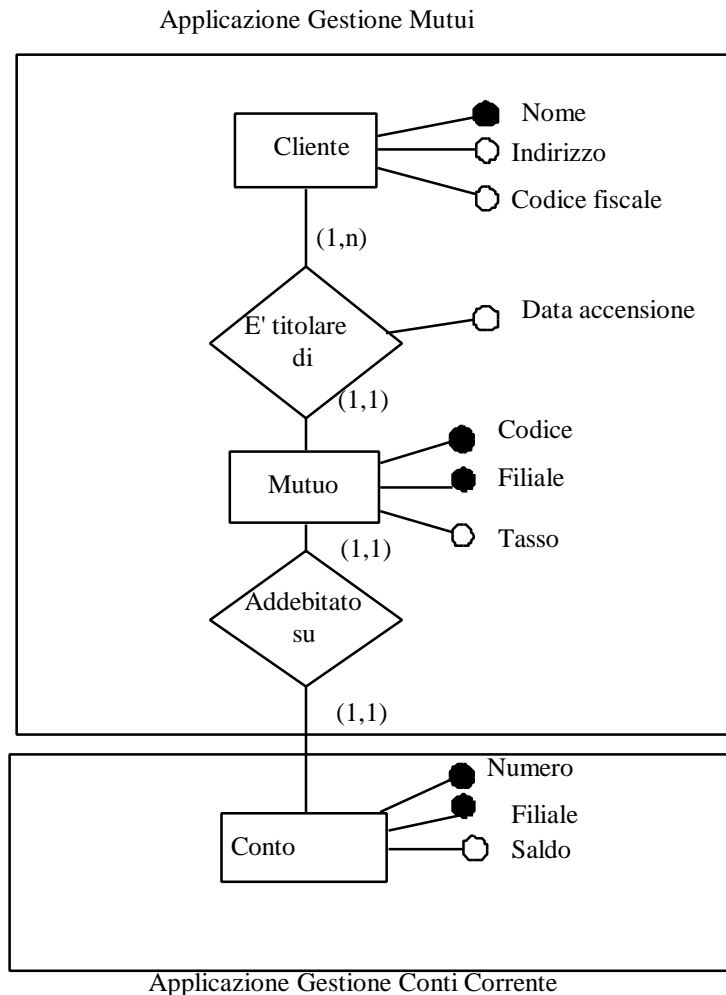
- a) Si trovino le espressioni regolari D-U per le variabili del seguente programma. Cosa suggerisce tale risultato?
- b) Si individui, inoltre, un'insieme di casi di test di dimensione minima che soddisfi il criterio di copertura dei comandi .

```
program ventitre
var X, Y, Z: integer;
begin
    read(X);
    read(Y);
    if X >= Y
        then Z := X - Y
        else Y := Y - X;
    while Y > 0 do
        begin
            Y := Y - 1;
            Z := Y - X
        end;
    if Z >= 0
        then Z := X + Y + Z;
    write(Z)
end.
```

PUNTI 6

Esercizio n. 5

Si consideri l'applicazione Gestione Mutui di una banca. L'organizzazione dei dati è rappresentata dal seguente diagramma ER:



Le linee tratteggiate indicano i confini dell'applicazione Gestione Mutui, che si deve contare, e dell'applicazione Gestione Conti Correnti.

L'applicazione Gestione Mutui deve eseguire le seguenti operazioni:

- 1 Gestione dei dati sui Clienti
 - 1.1 Inserisci Cliente: inserisci i dati di un nuovo cliente che ha chiesto un mutuo. I dati comprendono il Nome, l'Indirizzo e il CodiceFiscale;
 - 1.2 Interroga clienti: mostra la lista di tutti i nominativi che hanno chiesto un mutuo.
- 2 Gestione dei dati sui Mutui
 - 2.1 Inserisci Mutuo: inserisci i dati di un nuovo mutuo. I dati comprendono il Nome del Cliente, la Data di accensione del Mutuo, il Tasso praticato, il codice della Filiale e il Codice identificativo del mutuo, il numero di Conto Corrente sul quale viene addebitato e la Filiale del conto.
 - 2.2 Interroga Mutui per Cliente: dato il nome di un cliente, stampa i dati (Data accensione, codice e filiale, tasso) di tutti i mutui del cliente;

Si richiede di identificare le funzioni di tipo dati e di tipo transazione e di calcolare la complessità delle funzioni di tipo dati. Si calcoli, inoltre, la complessità della funzione Interroga Mutui per Cliente.

Si tenga presente che Conto è considerato come ILF per l'Applicazione Gestione Conti Correnti.

PUNTI 9