

Compito Scritto di Ingegneria del Software

15 Gennaio 1999 - h. 10 Aula 62.9

Tempo a disposizione: 3 ore

Esercizio n. 1

Modellare con una rete di Petri un sistema di controllo degli accessi a un laboratorio analisi. L'accesso avviene attraverso una porta solitamente chiusa e richiede il riconoscimento della persona tramite un tesserino magnetico. Un lettore di tessera magnetica provvede a questo scopo.

Quando la tessera viene inserita nel lettore, si apre la porta. L'ingresso della persona è rilevato da una cellula fotoelettrica. Quando la cellula rileva il passaggio, la porta si richiude.

PUNTI 5

Esercizio n. 2

Modellare in TRIO il funzionamento di una porta che si apre quando viene inserita una tessera magnetica in un apposito lettore e si richiude dopo trenta secondi.

PUNTI 2

Esercizio n. 3

Un laboratorio analisi di un ospedale ha quattro macchinari identici, ciascuno in grado di eseguire venti diversi tipi di analisi (identificate con un numero da 1 a 20). Le analisi vengono richieste dai medici dei vari reparti dell'ospedale. Ciascun medico è identificato da un codice, assegnato all'atto dell'assunzione del medico. I macchinari sono disponibili dalle 8 del mattino sino alle 19 e si suppone che ciascuna analisi richiesta duri un'ora.

Quando un medico prenota un'analisi, deve specificare il proprio codice e il tipo di analisi (da 1 a 20). Accertato che il medico appartenga all'ospedale, gli viene assegnato un orario (il più presto possibile) e un macchinario. Non c'è limite al numero di richieste che un medico può fare nel corso della giornata.

Un macchinario non può svolgere più di un'analisi allo stesso tempo.

La prenotazione va a buon fine solo se esiste un macchinario libero su cui svolgere l'analisi.

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 analisi.

PUNTI 8

Esercizio n. 4

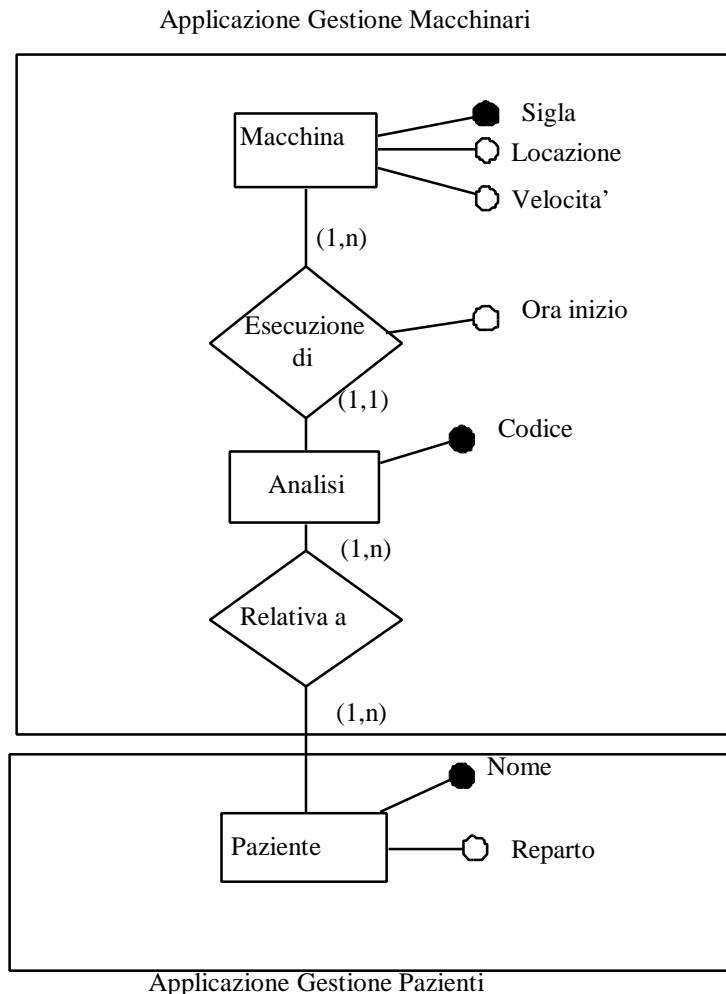
Si trovino le espressioni regolari D-U per le variabili del seguente programma. Cosa suggerisce il risultato?

```
program quindici
var A,B,C: integer;
begin
    read(A);
    read(B);
    if A>B
        then C=3*A+B
        else B=B-A;
    while B > 0 do
        begin
            B:= B-A;
            C = C - A;
        end;
    if C >= 0
        then write(Z)
end.
```

PUNTI 5

Esercizio n. 5

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



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

L'applicazione Gestione Macchinari deve eseguire le seguenti operazioni:

- 1 Gestione dei dati sui Macchinari
 - 1.1 Inserisci Macchina: inserisci i dati di una nuova macchina. I dati comprendono il Sigla della macchina, la sua Locazione e Velocità;
 - 1.2 Interroga Macchinari: mostra la lista di tutti i macchinari con data Velocità;
- 2 Gestione dei dati sulle Analisi
 - 2.1 Inserisci Richiesta di Analisi: inserisci i dati di una analisi richiesta per un paziente. I dati comprendono il codice dell'analisi, la sigla della macchina su cui verrà eseguita, l'ora di inizio e il Nome del paziente;
 - 2.2 Interroga Richiesta di Analisi per Paziente: dato il nome di un paziente, stampa i dati (codice analisi, sigla macchina e ora di inizio) di tutte le analisi relative a quel paziente.

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 Richiesta di Analisi per Paziente.

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

PUNTI 10

Compito Scritto di Ingegneria del Software
19 Febbraio 1999 - h. 10 Aula 62.9
Tempo a disposizione: 3 ore

Esercizio n. 1

Un allarme antincendio è costituito da un sensore termico, una sirena e una chiave. La sirena si attiva (a chiave disinserita) quando il sensore rileva una temperatura superiore ai 40 gradi e si disattiva quando viene inserita la chiave. . Modellare il sistema con una rete di Petri.

PUNTI 5

Esercizio n. 2

Modellare in TRIO il funzionamento di un allarme che si attiva quando un sensore rileva una temperatura superiore ai 40 gradi e si disattiva dopo tre minuti.

PUNTI 2

Esercizio n. 3

Si vuole specificare in Z la gestione di videocassette e di clienti (registrati) di un negozio. Ciascun utente è caratterizzato da nome e cognome e identificato da un codice che gli viene assegnato all'atto della registrazione. Si suppone che non esistano omonimi.

Esiste un elenco delle videocassette, ciascuna caratterizzata da un titolo e da un numero di inventario. Si suppone che esista al più una copia di ogni videocassetta, per semplicità. Una videocassetta può essere disponibile, oppure in prestito.

Si modellino le seguenti operazioni:

- 1) Registrazione di un nuovo cliente: dati nome e cognome del cliente, se non esiste alcun omonimo, viene assegnato un nuovo codice identificativo e si aggiunge il cliente all'elenco di quelli già registrati; altrimenti si dà una segnalazione di errore;
- 2) Richiesta di un prestito: dati il codice del cliente e il numero di inventario della videocassetta richiesta, il sistema controlla che esista un cliente in elenco con quel codice e una videocassetta disponibile con quel codice. In questo caso modifica lo stato della videocassetta da disponibile a "in prestito".

Inoltre il sistema deve far sì che NON sia possibile prestare videocassette a utenti con codice non registrato o prestare videocassette con numero di inventario non esistente.

PUNTI 8

Esercizio n. 3

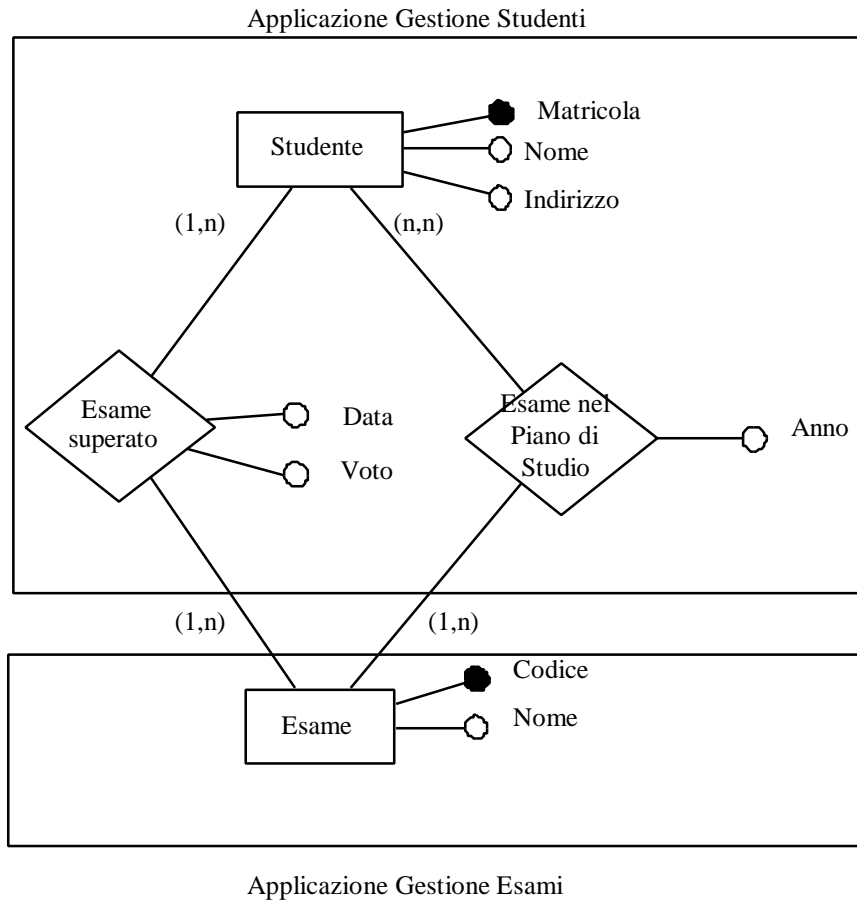
Si disegni il grafo di controllo per il programma seguente e si individui un'insieme di casi di test di dimensione minima che soddisfi il criterio di copertura dei comandi, commentando i risultati.

```
program Febbraio;
var A,B,C,D : real;
var Cond_A, Cond_B : boolean;
begin
  Cond_A:= true;
  Cond_B:= false;
  read(A);
  if A < 0
    then
      begin
        Cond_A := false;
        C := -A
      end
    else      C := A;
  read(B);
  if B > 0
    then
      Cond_B := true;
  while A <= B do
    begin
      C := C+1;
      if Cond_A
        then A:=A-1
        else A:=A+1;
      end;
  if not (Cond_A or Cond_B )
  then D := -C
  else D := C;
  write(C);
  write(D);
end.
```

PUNTI 6

Esercizio n. 4

Si consideri l'applicazione Gestione Studenti. L'organizzazione dei dati è rappresentata dal seguente diagramma ER:



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

L'applicazione Gestione Studenti deve eseguire le seguenti operazioni:

- 1 Gestione dei dati sugli Studenti
 - 1.1 Iscrizione Studente: inserisci i dati di un nuovo studente. I dati comprendono la Matricola, il Nome e l'Indirizzo..
 - 1.2 Interroga Studenti per Nome: richiedi i dati di uno studente dato il suo Nome.
 - 1.3 Interroga Studenti per Matricola: richiedi i dati di uno studente data la sua Matricola.
- 2 Gestione dei Piani di Studio
 - 2.1 Inserisci Piano di Studio: inserisci il piano di studio per uno studente. L'utente deve inserire la matricola dello studente e l'elenco degli esami. Per ogni esame, deve specificare il Codice e l'anno di corso in cui lo studente intende sostenere l'esame.
- 3 Gestione degli esami superati
 - 3.1 Inserisci Esame superato: inserisce i dati di un esame superato da uno studente. L'utente inserirà il Codice dell'esame superato, il codice dello Studente, la Data e il Voto.
 - 3.2 Interroga Esami superati: richiede gli esami superati da uno studente, data la sua matricola e stampa il numero totale di esami superati.

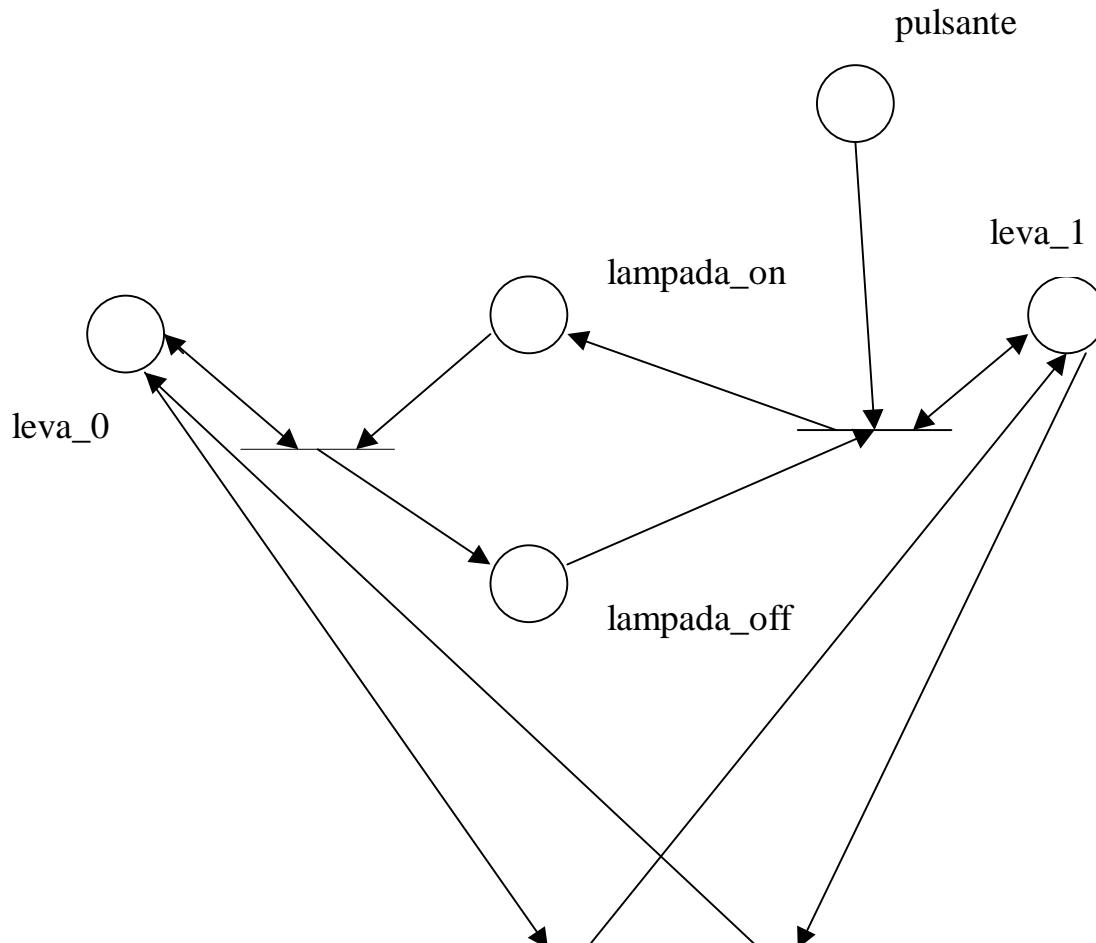
Si richiede di identificare le funzioni di tipo dati e di tipo transazione e di calcolare la complessità delle funzioni di tipo dati. Si tenga presente che Esami è considerato come ILF per l'Applicazione Gestione Esami.

I^o Compito Intermedio di Ingegneria del Software

21 Aprile 1999 - h. 16 aula E1B - tempo a disposizione 2 ore

Esercizio 1

Specificare mediante una rete di Petri il funzionamento di sistema composto da un interruttore a pressione, una leva a due posizioni (0,1) e una lampadina. La lampada si accende, da spenta che era, quando viene premuto il pulsante e la leva si trova in posizione 1. L'accensione della lampada lascia la leva nel suo stato. La lampada si spegne dopo che la leva viene girata e portata in posizione 0.



Esercizio 2

Specificare mediante TRIO il comportamento di un sistema costituito da un pulsante, una leva a due posizioni (alta e bassa) e una lampadina. La lampada si accende quando viene premuto il pulsante e la leva si trova nella posizione alta; la lampada si spegne dopo tre minuti al massimo oppure quando viene la leva viene portata nella posizione opposta.

Linguaggio:

pulsante, vero se pulsante premuto

lampada, vero se lampada accesa

leva, vero se leva alta

Unità di tempo: minuti

La seguente specifica TRIO cattura il comportamento di un tale sistema:

pulsante $\dot{\cup}$ leva $\dot{\cup}$ \emptyset lampada \ll Becomes(lampada)

\emptyset leva $\dot{\cup}$ Lasted(lampada,3) \ll Becomes(\emptyset lampada)

oppure si poteva usare Until:

pulsante $\dot{\cup}$ leva \rightarrow Until (lampada, \neg leva)

Lasted(lampada,3) \otimes \emptyset lampada

NB Until richiede che dopo l'accensione (leva alta)

si verifichi prima o poi spegnimento (leva bassa)

\Rightarrow la lampada non viene lasciata accesa indefinitamente.

Comunque, dopo che la lampada è stata accesa per 3 minuti, viene spenta.

Esercizio 4

Spiegare a cosa si possono applicare e a cosa possono servire le dichiarazioni *friend* all'interno di una classe C++.

La dichiarazione friend (che compare all'interno di una classe C) si applica a funzioni e a intere classi. In questo secondo caso, tutti i metodi di queste classi risultano "friend" della classe C.

La parola chiave friend posta all'interno di una classe C indica che la funzione (o tutti i metodi della classe indicata come friend) possono accedere agli elementi (dati e funzioni) privati (ovvero sia private sia protected) della classe C.

La relazione stabilita non è una relazione transitiva, ovvero se C dichiara friend C1 e C1 friend C2, C2 non è friend di C.

II Scritto Intermedio di Ingegneria del Software

26 Maggio 1999 – h. 16 Aula 3.3 – Tempo a disposizione 1h e 30'

Esercizio n. 1

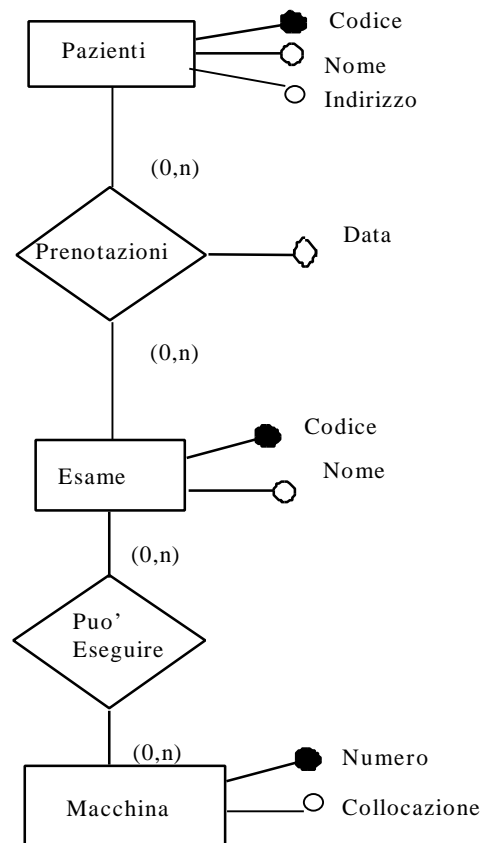
Dato il seguente programma, si determini l'insieme minimo di *cammini* da coprire selezionati dal criterio di copertura delle definizioni.

Dire come dovrebbe variare tale insieme per soddisfare il criterio di copertura di tutti gli usi (non è necessario indicare i cammini)

1	program ventisei;
2	var A, B:real;
3	X: real;
4	Cond: boolean;
5	begin
6	Cond:=false;
7	Read(A);
8	Read(B);
9	if (A>B)
10	then begin
11	X:=(A-B)/B;
12	Cond:=true;
13	end
14	else X:=(B-A)/B;
15	while Cond
16	do begin
17	A:= A-B;
18	if A<=0
19	then Cond:=false;
20	end
21	end.

PUNTI 5

Esercizio 2 Si consideri l'applicazione Gestione Laboratorio. L'organizzazione dei dati è rappresentata dal seguente diagramma ER:



L'applicazione deve eseguire le seguenti operazioni:

1. Gestione dei dati Pazienti
 - 1.1. Inserisci Paziente: inserisce i dati di un nuovo paziente. I dati comprendono il Nome, il Codice e l'Indirizzo.
 - 1.2. Interroga Pazienti: mostra i dati di un Paziente dato il suo Codice o il suo Nome.
 - 1.3. Mostra Totale Pazienti: mostra il numero totale di Pazienti in archivio.
2. Gestione degli Esami
 - 2.1. Inserisci Esame: inserisce i dati di un tipo di Esame. I dati comprendono il Nome e un Codice identificativo;
3. Gestione Prenotazioni
 - 3.1. Inserisci Prenotazione: inserisce una Prenotazione. I dati comprendono il Codice del paziente, il Codice dell'esame e la Data della Prenotazione;
 - 3.2. Interroga Prenotazioni: specificato il Codice di un paziente, visualizza l'elenco delle prenotazioni che lo riguardano, visualizzando Codice del Paziente, Data della Prenotazione, Nome e Codice dell'Esame;
4. Gestione delle Macchine
 - 4.1. Inserisci Macchina: inserisce i dati di una nuova macchina. I dati comprendono il Numero identificativo e la Collocazione della macchina e il Codice degli esami che la macchina può eseguire.

Si richiede di identificare le funzioni di tipo dati e di tipo transazione e di calcolarne la complessità in termini di DET, RET e DET, FTR.

PUNTI 10

Compito Scritto di Ingegneria del Software

9 Settembre 1999 - h. 15 Aula 4.2

Tempo a disposizione: 2 ore e 1/2

Esercizio n. 1

Modellare con TRIO (in alternativa con una rete di Petri) in un sistema di controllo degli accessi a un laboratorio informatico. L'accesso avviene attraverso una porta solitamente chiusa e richiede il riconoscimento della persona tramite un tesserino magnetico. Un lettore di tessera magnetica provvede a questo scopo.

Quando la tessera viene inserita nel lettore, si apre la porta. La porta si richiude dopo 20 secondi.

PUNTI 5

Esercizio n. 2

Un laboratorio informatico di un'università ha 30 PC identici. I PC sono disponibili dalle 8 del mattino sino alle 19 e si suppone che ciascuna sessione di lavoro duri multipli di un'ora.

Ogni studente dell'università è identificato da un tesserino magnetico che riporta un codice univoco intero.

Per accedere al laboratorio lo studente presenta il tesserino e prenota un PC per un orario e per una durata di un'ora al massimo. Se esiste un PC libero per viene assegnato allo studente. Un PC è assegnato a uno studente al massimo per volta.

Uno studente può fare al più due prenotazioni nell'arco di un giorno.

Si suppone che le prenotazioni siano effettuabili per i giorni feriali (lun-ven) della settimana.

Per semplicità si ipotizzi che:

- 1) Le prenotazioni possano essere effettuate solo per la settimana corrente;
- 2) L'ora sia individuata da un intero (compreso fra 8 e 18).

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

PUNTI 10

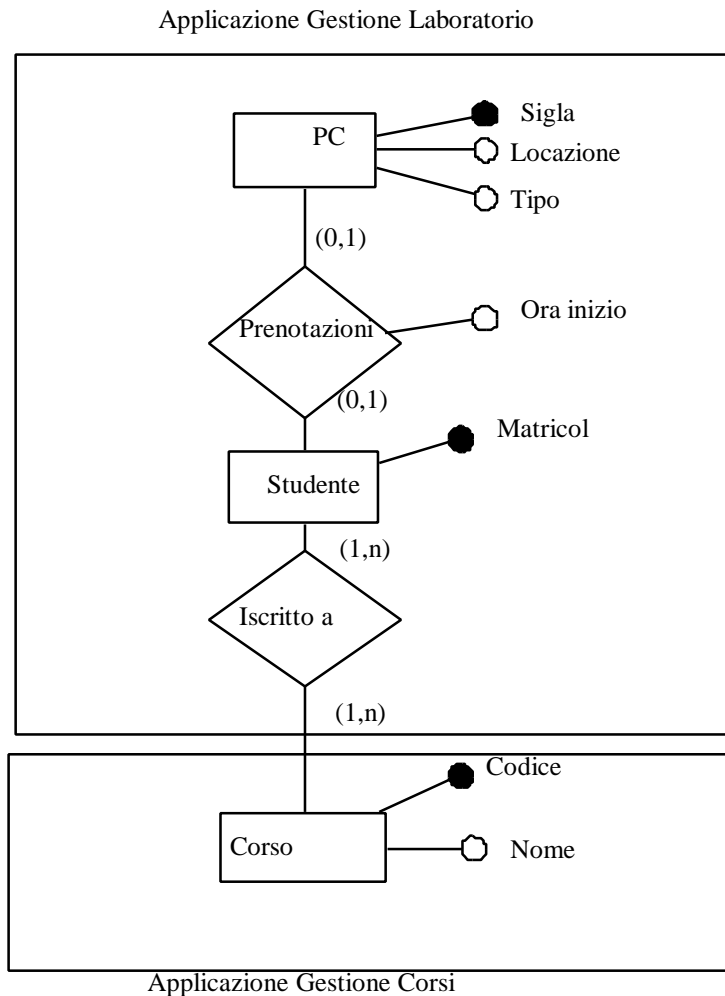
Esercizio n. 3

Si trovino le espressioni regolari D-U per le variabili del seguente programma. Cosa suggerisce il risultato?

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

Esercizio n. 4

Si consideri l'applicazione Gestione Laboratorio di un laboratorio informatico di un'università. L'organizzazione dei dati è rappresentata dal seguente diagramma ER:



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

L'applicazione Gestione Laboratorio deve eseguire le seguenti operazioni:

- 1 Gestione dei dati sui PC
 - 1.3 Inserisci PC: inserisci i dati di una nuova macchina. I dati comprendono un codice Sigla della macchina, la sua Locazione e Tipo;
 - 1.4 Interroga PC: mostra la lista di tutti i PC di un certo Tipo;
- 2 Gestione dei dati sulle Prenotazioni
 - 2.3 Inserisci Prenotazione: inserisci i dati della prenotazione di un PC da parte di uno studente. I dati comprendono la Matricola dello Studente, la Sigla del PC e l'ora di inizio della prenotazione;
- 3 Gestione dei dati sulle Iscrizioni a corsi
 - 3.1 Inserisci Iscritto: iscrivi uno studente a un corso. I dati comprendono la Matricola dello Studente, e il Codice del Corso.
 - 3.2 Mostra Totale Iscritti: mostra il numero totale degli iscritti a un corso, dato il Codice del Corso.

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

PUNTI 10

Scritto di Ingegneria del Software

5 Novembre 1999- h. 15 Aula 5.1 (durata 3 ore)

Esercizio 1

Specificare mediante una rete di Petri il funzionamento di un sistema di allarme di un edificio dotato di una sirena, tre sensori identici che rilevano presenze all'interno dell'edificio e una chiave di disinserimento dell'allarme. La sirena si accende quando almeno due dei tre sensori rilevano una presenza.

Dopo che è scattata la sirena, questa viene spenta se viene girata la chiave.

PUNTI 4

Esercizio n. 2

Un sistema di allarme è costituito da un sensore, una sirena e una leva a due posizioni (on/off). Il sistema può essere inserito (leva on) o disinserito (leva off).

Si scriva la specifica TRIO della seguente porzione di sistema:

- 1) A sistema inserito, la sirena si accende dopo 5 sec. dal rilevamento di una presenza da parte del sensore;
- 2) La sirena si spegne quando viene girata su off la leva.

PUNTI 3

Esercizio 3

Si dia una specifica in Z di una stazione di prenotazione esami di un ospedale.

Ciascun tipo di esame è dotato di un codice identificativo (alfanumerico). Ciascun utente è dotato di un codice identificativo (intero positivo). Esiste un limite massimo di prenotazioni per ciascun tipo di esami effettuabili nella singola giornata a una data ora (per semplicità supponiamo non più di 5 esami dello stesso tipo alla stessa ora).

Una prenotazione avviene inserendo il codice del tipo di esame, il codice dell'utente, la data e ora dell'esame.

La prenotazione va a buon fine se per quel tipo di esame nell'ora specificata e nel giorno indicato non è stato superato il limite delle 5 prenotazioni. **Un utente può effettuare una sola prenotazione a una data ora e giorno.**

Una prenotazione può essere cancellata specificando il codice del paziente, l'ora e la data.

Si chiede di specificare le seguenti operazioni in Z:

- inserimento di una prenotazione;
- cancellazione di una prenotazione.

PUNTI 9

Esercizio n. 4

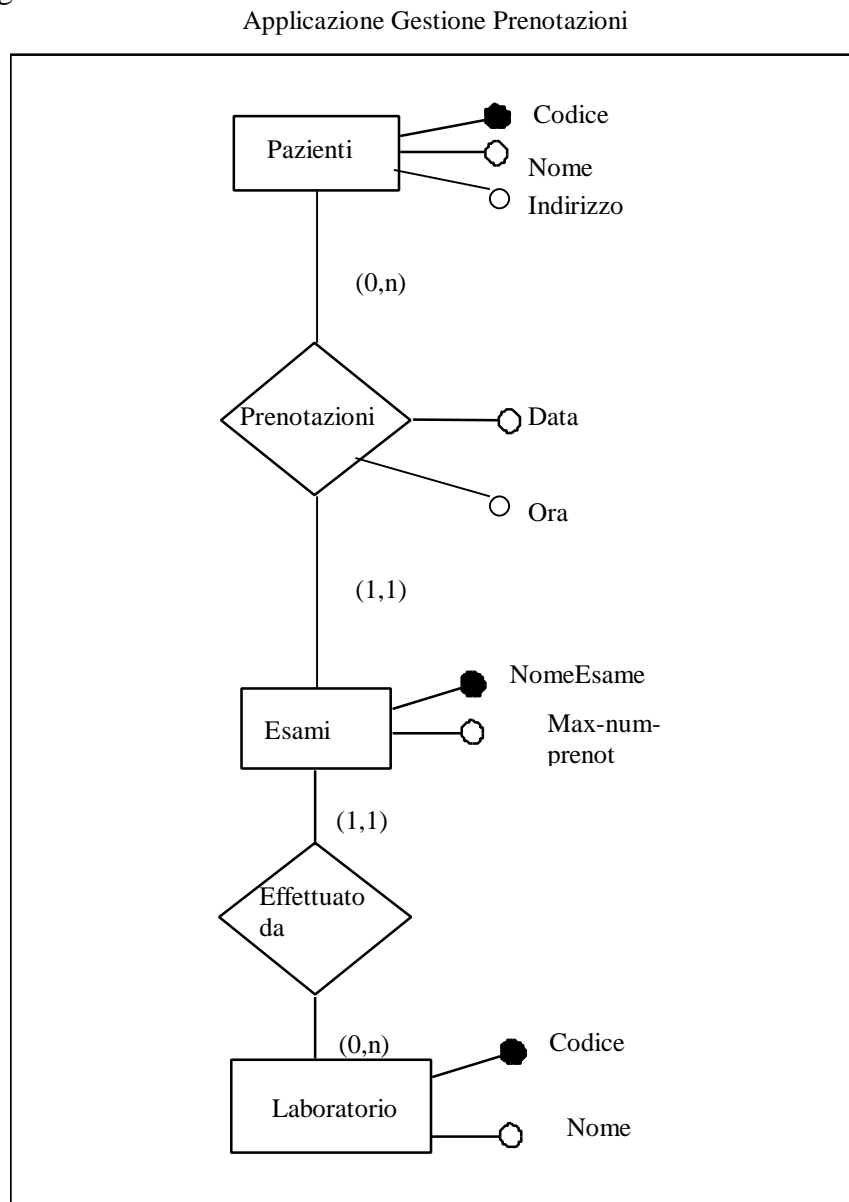
Si individui un insieme di casi di test di dimensione minima che soddisfi il criterio di copertura delle decisioni per il programma seguente:

```
program novembre;  
var X,Y: real;  
var Cond1: boolean;  
begin  
    Cond1 := false;  
    read(X);  
    read(Y);  
    if X>=Y then Cond1:= true;  
    if Cond1  
    then  
        while X>=Y do  
            X := X-1;  
        else while Y>X do  
            Y := Y-1;  
    write("fine");  
end.
```

PUNTI 5

Esercizio 5

Si consideri l'applicazione Gestione Prenotazione Esami. L'organizzazione dei dati è rappresentata dal seguente diagramma ER:



L'applicazione deve eseguire le seguenti operazioni:

5. Gestione dei dati Pazienti

5.1. Inserisci Paziente: inserisci i dati di un nuovo paziente. I dati comprendono il Nome, il Codice e l'Indirizzo.

6. Gestione dei dati degli Esami

6.1. Inserisci Esame: inserisci i dati di un nuovo esame. I dati comprendono il Nome dell'Esame e il NumeroMassimo di Prenotazioni per quell'Esame.

7. Gestione dei dati Laboratori

7.1. Inserisci Laboratorio: inserisci i dati di un nuovo laboratorio convenzionato. I dati comprendono il Nome e il Codice e il Nome degli Esami che il laboratorio può effettuare (tale lista di nomi può anche essere vuota).

7.2. Interroga Laboratori: richiedi tutti i dati di un laboratorio (compreso l'elenco di Nomi degli Esami che può effettuare) dato il suo Codice.

8. Gestione dei dati sulle Prenotazioni

8.1. Inserisci Prenotazione: inserisci i dati di una prenotazione. I dati comprendono il Nome dell'Esame, la Data, l'Ora e il Codice del paziente che deve effettuare l'esame;

8.2. Mostra Prenotazioni: mostra di ciascuna Prenotazione il Nome dell'Esame, la Data, e l'Ora.

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

PUNTI 9

Compito Scritto di Ingegneria del Software

24 Febbraio 2000 – h. 15 Aula 5.4

Tempo a disposizione: 3 ore

Esercizio n. 1

Modellare con una rete di Petri un sistema di controllo costituito da un rubinetto comandato, un serbatoio e due sensori di livello. Il rubinetto viene aperto non appena primo il sensore rileva assenza di acqua e viene chiuso quando il secondo sensore rileva acqua.

PUNTI 5

Esercizio n. 3

Un laboratorio analisi di un ospedale ha nove macchinari identici, ciascuno in grado di eseguire venti diversi tipi di analisi (identificate con un numero da 1 a 100). Le analisi vengono richieste dai reparti dell'ospedale. Ciascun reparto è identificato da un codice (intero positivo da 1 a 10). I macchinari sono disponibili dalle 8 del mattino sino alle 14 e si suppone che ciascuna analisi richieda duri un'ora.

Quando si prenota un'analisi, si deve specificare il codice del reparto e il tipo di analisi. La prenotazione va a buon fine solo se esiste un macchinario libero su cui svolgere l'analisi. Accertata la disponibilità, viene assegnato un orario (il più presto possibile) e un macchinario.

Non c'è limite al numero di richieste che un reparto può fare nel corso della giornata.

Un macchinario non può svolgere più di un'analisi allo stesso tempo.

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 14).

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

PUNTI 9

Esercizio n. 4

Dato il seguente programma, si determinino un insieme minimo di *cammini* da coprire selezionato dal criterio di copertura di tutti gli usi.

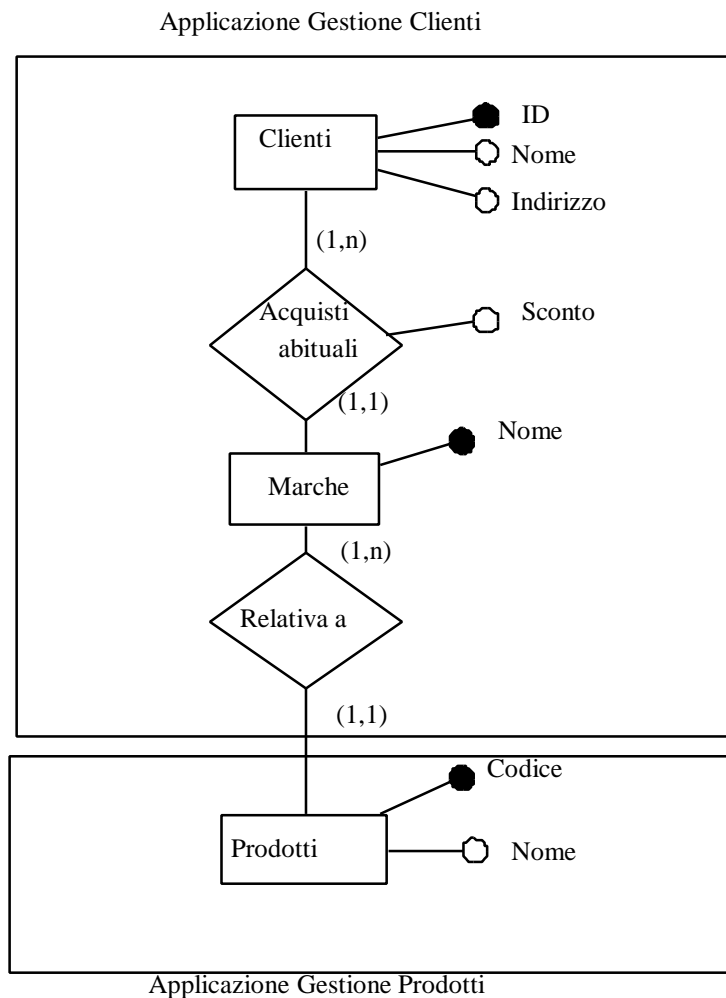
1	program primo;
2	var A,B: Integer;
3	X: real;
4	T: boolean;
5	begin
6	T:=false;
7	Read(A);
8	Read(B);
9	if A>B
10	then begin
11	X:=(A-B)/A;
12	T:=true;
13	end
14	else X:=(B-A)/B;
15	while T
16	do begin
17	A:= A-B;
18	if A<=0

19	then T:=false;
20	end
21	end.

PUNTI 6

Esercizio n. 5

Si consideri l'applicazione Gestione Clienti. L'organizzazione dei dati è rappresentata dal seguente diagramma ER:



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

L'applicazione Gestione Clienti deve eseguire le seguenti operazioni:

- 1 Gestione dei dati sui Clienti
 - 1.5 Inserisci Cliente: inserisci i dati di un nuovo cliente. I dati comprendono il Nome, l'Indirizzo e un Identificativo progressivo e tutti gli Sconti attribuiti ai clienti per le varie Marche di prodotti;
 - 1.6 Interroga Clienti: mostra la lista di tutti i Clienti, indicando per ciascuno il Nome, l'Indirizzo e per ogni singola Marca lo Sconto applicabile al Cliente;
- 2 Gestione dei dati sulle Marche
 - 3.3 Inserisci Marca: inserisci i dati di una marca di prodotti, indicando il suo Nome e il Codice dei Prodotti in listino ai quali si riferisce;
 - 3.4 Interroga Marche per Prodotto: dato il Nome di una Marca di Prodotti, riporta in uscita il numero totale di Prodotti in listino per quella Marca;

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 Clienti.

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

PUNTI 10

I^o Compito Intermedio di Ingegneria del Software - A.A. 99/00

27 Aprile 2000 - h. 15 aula 6.2 (durata 2h – TOT 20 punti)

Esercizio 1

Un sistema software è costituito da un processo produttore e un processo consumatore, che si scambiano informazioni mediante un buffer (capace di memorizzare un solo messaggio). Inizialmente, il produttore è pronto a scrivere e il consumatore è pronto a leggere un messaggio dal buffer. Il produttore può scrivere sul buffer solo se il buffer è libero. Dopo che ha scritto sul buffer, il produttore attende un messaggio di ricevuto che viene inviato dal consumatore (su un canale di comunicazione diverso) alla lettura del messaggio e solo a questo punto può passare nello stato di pronto a produrre un nuovo messaggio.

Specificare mediante una rete di Petri il funzionamento di tale sistema. PUNTI 5

Esercizio 2

Si dia una specifica in Z del sistema informativo di un centro termale privato che eroga ai suoi iscritti sei diversi tipi di corso nelle sue tre piscine. I corsi sono tenuti dalle 12 alle 17 e ciascun corso occupa l'ora successiva il suo inizio.

Per ogni piscina, nell'ora stabilita e a seconda del tipo di corso, è definito il numero massimo di persone che possono partecipare. Si supponga che questa informazione sia rappresentata nel database dalla funzione:

$$n_{\max}: \text{Num_Piscina} \times \text{Num_Ora} \times \text{Num_Corso} \rightarrow N_+$$

dove il primo elemento del dominio della funzione rappresenta il numero della piscina (da 1 a 3), il secondo l'ora (dalle 12 ora 0, alle 17 ora 5) e il terzo il tipo di corso (da 1 a 6). Se per esempio, nella prima piscina alle 12 si tiene il corso di acquagym (per esempio, il quarto dei sei) con numero massimo consentito di partecipanti uguale a 45, si avrà per la funzione:

$$n_{\max}(1,0,4)=45$$

Le tuple della funzione n_{\max} siano date per assegnate nel database.

Ciascuna persona che vuole accedere ai corsi si deve *iscrivere* al centro, indicando il codice fiscale, il nome e la data di nascita. All'iscrizione viene controllato che non esista già una persona in archivio con lo stesso codice fiscale e se l'iscrizione va a buon fine viene rilasciato un tesserino magnetico (recante come identificativo il codice fiscale).

All'arrivo in piscina, tramite la lettura del tesserino magnetico, viene *registrato* l'ingresso della persona.

Solo dopo l'ingresso, la persona può *prenotarsi* per uno o più corsi (richiamando più volte l'operazione di prenotazione). Per effettuare ciascuna prenotazione la persona deve inserire il tesserino in un lettore e indicare il corso e l'ora in cui desidera seguirlo. La prenotazione va a buon fine se esiste il corso indicato per quell'ora e se non è già stato raggiunto il numero massimo di partecipanti (in termini di prenotati). L'operazione di prenotazione che va a buon fine restituisce in uscita il numero della piscina nella quale si tiene il corso all'ora indicata.

Il numero di prenotati per qualunque corso, ora e piscina è azzerato automaticamente a fine

giornata.

Si dia la specifica in Z delle seguente operazioni:

- a) Iscrizione(Nome,DataNascita,CF): inserisce un nuovo iscritto. L'operazione va a buon fine se non esiste già un iscritto con lo stesso codice fiscale.
- b) Ingresso(CF): registra l'ingresso di un iscritto all'interno del centro. L'operazione va a buon fine se esiste un **iscritto** con il codice fiscale (CF) uguale a quello letto dal tesserino magnetico e in tal caso la persona viene aggiunta all'insieme dei **presenti** in piscina.
- c) Prenota(CF,Ora,Numero_Corso): prenotazione di un corso da parte di un presente all'interno della piscina. L'operazione va a buon fine se la persona con CF specificato risulta tra i **presenti** e se esiste nel database dei corsi (rappresentato dalla funzione n_{max}) un corso con il Numero_Corso indicato all'Ora indicata. In tal caso, l'operazione restituisce in uscita il numero della piscina che ospita il corso all'ora specificata (si suppone ci sia al più una sola piscina possibile). PUNTI 8

Esercizio 3

Si determini un insieme minimo di **cammini** sufficienti a soddisfare il criterio di tutti gli usi (nota bene: non limitarsi a trovare i cammini D-U, ma individuare i cammini che corrispondono a esecuzioni complete).

1	program esercizio-tre
2	var A,B,C: integer
3	begin
4	Read(A);
5	Read(B);
6	if A>=B
7	then C := A-B
8	else C := B-A;
9	while A > 0 do
10	begin
11	A:= A-1;
12	C:= C-A
13	end;
14	if C > 0
15	then WRITE(B,C)
16	else WRITE(A,C)
17	end

PUNTI 7