

Corso di Programmazione

III Accertamento del 18 Giugno 2001 / A

cognome e nome

Risolvi i seguenti esercizi, riporta le soluzioni in modo chiaro negli appositi riquadri e giustifica sinteticamente le risposte utilizzando i fogli protocollo.

1. Procedure in Scheme

Definisci una procedura in Scheme che, dato un vettore, ne rovescia l'ordine degli elementi senza duplicarlo. In altri termini, dopo l'esecuzione della procedura la configurazione del vettore passato come parametro risulta "speculare" rispetto alla configurazione prima dell'esecuzione.

2. Modello funzionale di un programma iterativo e invarianti

Definisci un programma in Scheme, basato sulla ricorsione, che rappresenti il modello funzionale del seguente programma iterativo in Java:

```
// x, y naturali positivi

    z = y;
    while ((z % x) != 0) {
        z = z + y;
    }

// z è il minimo comune multiplo di x e y
```

A tale fine si utilizzi una rappresentazione M della memoria definita (in Scheme) come segue:

```
(define x 0)
(define y 1)
(define z 2)
(define M (make-vector 3))
(vector-set! M x <un certo valore naturale positivo> )
(vector-set! M y <un altro valore naturale positivo> )
```

Data una memoria *iniziale* definita come sopra, il valore assunto dalla procedura principale del programma deve rappresentare la memoria *alla fine* dell'esecuzione del programma iterativo in Java.

Inoltre, descrivi sinteticamente (in modo informale) quali potrebbero essere le asserzioni invarianti del comando *while* del programma in Java che ne dimostrano la correttezza.

3. Programmi iterativi in Java

Definisci un programma in Java che risolva lo stesso problema proposto dall'esercizio n. 1.

4. Classi in Java

Considera la seguente versione del gioco del Nim: all'inizio ci sono n scatole, contenenti rispettivamente n monete, $n-1$ monete, ..., 2 monete, 1 moneta. I due giocatori, a turno, possono prelevare un numero di monete (delle monete rimaste) compreso fra 1 e m da una sola delle n scatole, dove m è una costante prefissata. Perde chi preleva l'ultima moneta.

Definisci una classe in Java che rappresenti lo stato del gioco. Tale classe deve prevedere un costruttore con parametro n per rappresentare la configurazione iniziale, un metodo per realizzare la mossa di un giocatore che intende prelevare k monete dall' i -ima scatola (segnalando se la mossa non è lecita) e un metodo per verificare se il giocatore di turno ha perso.

Corso di Programmazione

III Accertamento del 18 Giugno 2001 / B

cognome e nome

Risolvi i seguenti esercizi, riporta le soluzioni in modo chiaro negli appositi riquadri e giustifica sinteticamente le risposte utilizzando i fogli protocollo.

1. Procedure in Scheme

Definisci una procedura in Scheme che, dato un vettore, operi una rotazione di una posizione a sinistra senza duplicarlo. In altri termini, dopo l'esecuzione della procedura ciascun elemento si trova spostato a sinistra di una posizione, ad eccezione del primo che invece passa in ultima posizione rispetto alla configurazione del vettore prima dell'esecuzione.

2. Modello funzionale di un programma iterativo e invarianti

Definisci un programma in Scheme, basato sulla ricorsione, che rappresenti il modello funzionale del seguente programma iterativo in Java:

```
// n naturale positivo
    p = true;  k = 2;
    while (p & (k < n)) {
        p = ((n % k) != 0);  k = k+1;
    }

// p è vero se e solo se n è primo
```

A tale fine si utilizzi una rappresentazione M della memoria definita (in Scheme) come segue:

```
(define n 0)
(define p 1)
(define k 2)
(define M (make-vector 3))
(vector-set! M n <un certo valore naturale positivo> )
```

Data una memoria *iniziale* definita come sopra, il valore assunto dalla procedura principale del programma deve rappresentare la memoria *alla fine* dell'esecuzione del programma iterativo in Java.

Inoltre, descrivi sinteticamente (in modo informale) quali potrebbero essere le asserzioni invarianti del comando *while* del programma in Java che ne dimostrano la correttezza.

3. Programmi iterativi in Java

Definisci un programma in Java che risolva lo stesso problema proposto dall'esercizio n. 1.

4. Classi in Java

Considera la seguente versione del gioco del Nim: all'inizio ci sono n scatole, contenenti rispettivamente $2n-1$ monete, $2n-3$ monete, ..., 3 monete, 1 moneta. I due giocatori, a turno, possono prelevare un numero qualsiasi di monete (delle monete rimaste), e ne devono prelevare almeno una, da una sola delle n scatole. Perde chi preleva l'ultima moneta.

Definisci una classe in Java che rappresenti lo stato del gioco. Tale classe deve prevedere un costruttore con parametro n per rappresentare la configurazione iniziale, un metodo per realizzare la mossa di un giocatore che intende prelevare k monete dall' i -ima scatola (segnalando se la mossa non è lecita) e un metodo per verificare se il giocatore di turno ha perso.

Corso di Programmazione

III Accertamento del 18 Giugno 2001 / C

cognome e nome

Risolvi i seguenti esercizi, riporta le soluzioni in modo chiaro negli appositi riquadri e giustifica sinteticamente le risposte utilizzando i fogli protocollo.

1. Procedure in Scheme

Definisci una procedura in Scheme che, dato un vettore di interi, sostituisca il valore di ciascun elemento con la somma dell'elemento stesso e di tutti i precedenti (senza duplicare il vettore), dove si intendono i valori degli elementi prima dell'esecuzione della procedura.

2. Modello funzionale di un programma iterativo e invarianti

Definisci un programma in Scheme, basato sulla ricorsione, che rappresenti il modello funzionale del seguente programma iterativo in Java:

```
// x, y naturali positivi

z = y;
while ((x % z) != 0) | ((y % z) != 0) {
    z = z - 1;
}

// z è il massimo divisore comune di x e y
```

A tale fine si utilizzi una rappresentazione M della memoria definita (in Scheme) come segue:

```
(define x 0)
(define y 1)
(define z 2)
(define M (make-vector 3))
(vector-set! M x <un certo valore naturale positivo> )
(vector-set! M y <un altro valore naturale positivo> )
```

Data una memoria *iniziale* definita come sopra, il valore assunto dalla procedura principale del programma deve rappresentare la memoria *alla fine* dell'esecuzione del programma iterativo in Java.

Inoltre, descrivi sinteticamente (in modo informale) quali potrebbero essere le asserzioni invarianti del comando *while* del programma in Java che ne dimostrano la correttezza.

3. Programmi iterativi in Java

Definisci un programma in Java che risolva lo stesso problema proposto dall'esercizio n. 1.

4. Classi in Java

Considera la seguente versione del gioco del Nim: all'inizio ci sono n scatole, contenenti rispettivamente n monete, $n-1$ monete, ..., 2 monete, 1 moneta. I due giocatori, a turno, possono prelevare un numero di monete (delle monete rimaste) compreso fra 1 e m da una sola delle n scatole, dove m è una costante prefissata. Perde chi preleva l'ultima moneta.

Definisci una classe in Java che rappresenti lo stato del gioco. Tale classe deve prevedere un costruttore con parametro n per rappresentare la configurazione iniziale, un metodo per realizzare la mossa di un giocatore che intende prelevare k monete dall' i -ima scatola (segnalando se la mossa non è lecita) e un metodo per verificare se il giocatore di turno ha perso.

Corso di Programmazione

III Accertamento del 18 Giugno 2001 / D

cognome e nome

Risolvi i seguenti esercizi, riporta le soluzioni in modo chiaro negli appositi riquadri e giustifica sinteticamente le risposte utilizzando i fogli protocollo.

1. Procedure in Scheme

Definisci una procedura in Scheme che, dato un vettore, operi una rotazione di una posizione a destra senza duplicarlo. In altri termini, dopo l'esecuzione della procedura ciascun elemento si trova spostato a destra di una posizione, ad eccezione dell'ultimo che invece passa in prima posizione rispetto alla configurazione del vettore precedente all'esecuzione.

2. Modello funzionale di un programma iterativo e invarianti

Definisci un programma in Scheme, basato sulla ricorsione, che rappresenti il modello funzionale del seguente programma iterativo in Java:

```
// n naturale positivo
    p = true;  k = n-1;
    while (p & (k > 1)) {
        p = ((n % k) != 0);  k = k-1;
    }

// p è vero se e solo se n è primo
```

A tale fine si utilizzi una rappresentazione M della memoria definita (in Scheme) come segue:

```
(define n 0)
(define p 1)
(define k 2)
(define M (make-vector 3))
(vector-set! M n <un certo valore naturale positivo> )
```

Data una memoria *iniziale* definita come sopra, il valore assunto dalla procedura principale del programma deve rappresentare la memoria *alla fine* dell'esecuzione del programma iterativo in Java.

Inoltre, descrivi sinteticamente (in modo informale) quali potrebbero essere le asserzioni invarianti del comando *while* del programma in Java che ne dimostrano la correttezza.

3. Programmi iterativi in Java

Definisci un programma in Java che risolva lo stesso problema proposto dall'esercizio n. 1.

4. Classi in Java

Considera la seguente versione del gioco del Nim: all'inizio ci sono n scatole, contenenti rispettivamente $2n-1$ monete, $2n-3$ monete, ..., 3 monete, 1 moneta. I due giocatori, a turno, possono prelevare un numero qualsiasi di monete (delle monete rimaste), e ne devono prelevare almeno una, da una sola delle n scatole. Perde chi preleva l'ultima moneta.

Definisci una classe in Java che rappresenti lo stato del gioco. Tale classe deve prevedere un costruttore con parametro n per rappresentare la configurazione iniziale, un metodo per realizzare la mossa di un giocatore che intende prelevare k monete dall' i -ima scatola (segnalando se la mossa non è lecita) e un metodo per verificare se il giocatore di turno ha perso.