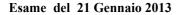
\sim		D	•
Corco	aı	Programn	ทรวเกทค
CUISU	uı	I I VEI ami	Huziviic



cognome e nome			

Risolvi i seguenti esercizi, riporta le soluzioni in modo chiaro negli appositi spazi e giustifica sinteticamente le risposte. Dovrai poi consegnare queste schede con le soluzioni, avendo cura di scrivere il tuo nome nell'intestazione e su ciascun eventuale foglio aggiuntivo.

1. Programmi in Java

Un numero x compreso nell'intervallo J-0.5, +0.5I (cioè tale che $|x| < \frac{1}{2}$) può essere codificato, almeno in forma approssimata, tramite una sequenza finita di cifre del sistema ternario bilanciato (-/./+), cifre che comparirebbero come parte frazionaria in questa notazione. Definisci in Java un metodo statico fractPart che, data una stringa btf di -/./+, restituisce il valore numerico x di btf, interpretata come parte frazionaria nel sistema ternario bilanciato. Esempi (i valori sono convenzionalmente approssimati alla terza cifra dopo il punto decimale):

```
fractPart("--") → -0.444 fractPart("--+") → -0.296 fractPart(".") → 0.000 fractPart(".++") → 0.148 fractPart("+--") → 0.296 fractPart("+") → 0.333
```

2. Programmi in Scheme

Facendo riferimento alla procedura tilings definita nella pagina seguente, determina il risultato della valutazione di ciascuna delle espressioni riportate qui sotto:

(tilings 1)	\rightarrow	 (tilings 4)	\rightarrow	
(tilings 2)	\rightarrow	 (tilings 6)	\rightarrow	
(tilings 3)	\rightarrow	 (tilings 8)	\rightarrow	

3. Procedure con valori procedurali

Il cifrario di Vigenère (1523–1596) rappresenta una generalizzazione del cifrario di Cesare, in base al quale la rotazione delle lettere dell'alfabeto non è costante, ma varia ciclicamente in funzione di una parola chiave, concordata fra mittente e destinatario, che rappresenta le successive "mappature" della lettera iniziale A. Il meccanismo si può illustrare più facilmente ricorrendo a un esempio. Supponiamo che la parola chiave sia "AKEY". Questo significa che la prima lettera del messaggio verrà ruotata di 0 posizioni ($A \rightarrow A$, $B \rightarrow B$, $C \rightarrow C$, ...); la seconda lettera del messaggio di 10 posizioni ($A \rightarrow K$, $B \rightarrow L$, $C \rightarrow M$, ...); la terza di 4 ($A \rightarrow E$, $B \rightarrow F$, $C \rightarrow G$, ...); la quarta di 24 ($A \rightarrow Y$, $B \rightarrow Z$, $C \rightarrow A$, ...); la quinta di nuovo di 0 posizioni ($A \rightarrow A$, ...) e così via ripetendo rotazioni della stessa entità con periodo 4 (= lunghezza della chiave). Quindi, a partire dal messaggio in chiaro "ASHORTSAMPLEMESSAGE" si può procedere secondo questo schema:

```
      A S H O R T S A M P L E M E S S A G E
      messaggio in chiaro

      A K E Y A K E Y A K E Y A K E Y A K E Y A K E M O W Q A Q I
      messaggio in chiaro

messaggio in chiaro
parola chiave ripetuta
messaggio crittato
```

Da cui risulta il testo crittato (terza riga): "ACLMRDWYMZPCMOWQAQI".

Data una parola chiave *key* (stringa di lettere maiuscole), la procedura vigenere-cipher restituisce la procedura di crittazione che applica il cifrario di Vigenère con chiave *key* per trasformare un messaggio in chiaro (stringa di lettere maiuscole) nel corrispondente testo crittato (stringa di lettere maiuscole). Per esempio:

```
(define encrypt (vigenere-cipher "AKEY")) (encrypt "ASHORTSAMPLEMESSAGE") \rightarrow "ACLMRDWYMZPCMOWOAOI"
```

Completa la procedura riportata qui sotto, che realizza vigenere-cipher, introducendo il codice Scheme appropriato negli spazi indicati a tratto punteggiato.

4. Verifica formale della correttezza

In relazione alla procedura definita sopra è possibile dimostrare che per tutte le coppie di valori interi $x, y \ge 0$:

$$(g \ x \ y) \rightarrow xy + |x-y|$$

Di	mostra per induzione questa proprietà; in particolare:
•	Formalizza la proprietà che esprime il caso / i casi base:
•	Formalizza l'ipotesi induttiva:
•	Formalizza la proprietà da dimostrare come passo induttivo:
•	Dimostra il caso / i casi base:
•	Dimostra il passo induttivo:

5. Memoization Trasforma il programma in Scheme dell'esercizio 2 in un programma in Java più efficiente, applicando opportunamente una tecnica top-down di *memoization*.