

Scrivete un programma che calcoli e^x usando questa formula. Ovviamente, non potete calcolare una sommatoria infinita: continuate semplicemente ad aggiungere termini finché un singolo addendo non sia minore di una determinata soglia. A ogni passo, dovete calcolare il nuovo termine e aggiungerlo al totale. Aggiornate tali termini come segue:

```
term = term * x / n;
```

Seguite l'approccio dei due esercizi precedenti, realizzando una classe `ExpApproximator`.

Esercizio P6.15. Scrivete un applet grafico che visualizzi una scacchiera con 64 caselle, alternativamente bianche e nere.

Esercizio P6.16. *Il gioco di Nim.* Si tratta di un gioco ben conosciuto, con un certo numero di varianti. Utilizzeremo questa versione, che ha una strategia interessante per arrivare alla vittoria. Due giocatori prelevano a turno biglie da un mucchio. In ciascun turno, il giocatore sceglie quante biglie prendere: deve prenderne almeno una, ma non oltre la metà del mucchio. Quindi tocca all'altro giocatore. Perde chi è costretto a prendere l'ultima biglia.

Scrivete un programma in cui il computer gioca contro un avversario umano. Generate un numero intero casuale, compreso fra 10 e 100, per indicare il numero iniziale di biglie. Generate un altro numero intero casuale, compreso fra zero e uno, per decidere se la prima mossa tocca al computer o al giocatore. Generate un numero intero casuale, compreso fra zero e uno, per stabilire se il computer giocherà in modo *intelligente* o *stupido*. Nel modo stupido, quando è il suo turno, il computer si limita a sottrarre dal mucchio un numero casuale di biglie, purché sia una quantità ammessa (compresa fra 1 e $n/2$). Nel modo intelligente, il computer leva il numero di biglie sufficiente affinché il numero di quelle rimanenti sia uguale a una potenza di due, meno uno, ovvero 3, 7, 15, 31 o 63. È sempre una mossa valida, eccetto quando il numero delle biglie è inferiore di un'unità a una potenza di due. In questo caso, il computer preleverà una quantità casuale, purché ammessa.

Noterete che, nel modo intelligente, il computer non si può battere quando ha la prima mossa, a meno che il mucchio non contenga 15, 31 o 63 biglie. Naturalmente, un giocatore umano che ha la prima mossa, e che conosca la strategia vincente, può vincere contro il computer.

Esercizio P6.17. Programmate la simulazione seguente. Si lanciano alcune frecce in punti casuali, all'interno di un quadrato con gli angoli nelle posizioni (1, 1) e (-1, -1). Si fa centro quando la freccetta cade all'interno del cerchio unitario (vale a dire la circonferenza con centro (0, 0) e con raggio uguale a 1). Altrimenti, è un colpo mancato. Eseguite la simulazione e adoperatela per stabilire un valore approssimato per π . Spiegate perché questo è un metodo migliore per stimare π , rispetto al programma dell'ago di Buffon.

Esercizio P6.18. Tracciare grafici di curve mediante la libreria grafica Java è facile e divertente. Disegnate semplicemente un centinaio di segmenti di linea per collegare i punti $(x, f(x))$ e $(x + d, f(x + d))$, dove x varia da x_{\min} a x_{\max} e dove d è uguale a $d = (x_{\max} - x_{\min}) / n$.