Fra parentesi

Alonzo ha scritto alcuni programmi per elaborare elenchi di carte da gioco (quelle di tipo francese, con un mazzo di 52 carte senza jolly). Le carte vengono ordinate in senso crescente secondo la sequenza A,2,3,...,10,J,Q,K e il seme picche ♠, fiori ♣, quadri ♢, cuori ♡: al K di picche, segue l'asso di fiori, e cosi` via fino al K di cuori che è quindi la carta più alta.

I programmi sono:

- (max x1 x2 x3 … xn) calcola la carta massima fra x1 x2 x3 … xn
- (min x1 x2 x3 … xn) calcola la carta minima fra x1 x2 x3 … xn
- (+ x1 x2 x3 … xn) calcola la somma fra i valori numerici di x1 x2 x3 … xn (A vale 1, J vale 11, Q vale 12 e K vale 13)
- (* x1 x2 x3 … xn) cambia il seme di ogni carta secondo questo schema: Cuori -> Quadri, Quadri -> Fiori, Fiori -> Picche, Picche -> Cuori

Le operazioni si possono anche annidare una nell'altra, per esempio

(+ (* 2♡ 3♣) (* (max A♠ 2♣)))

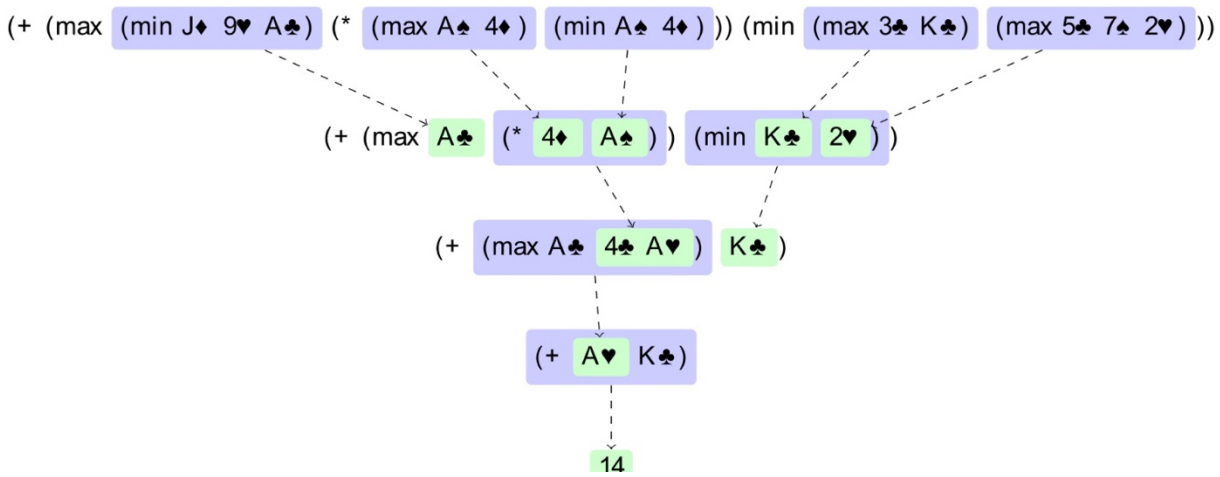
vale 7.

Che valore ha questa espressione?

(+ (max (min J♢ 9♡ A♣) (* (max A♠ 4♢) (min A♠ 4♢))) (min (max 3♣ K♣) (max 5♣ 7♠ 2♡)))

- Spiegazione -

Il valore dell'espressione data è **14**. Il problema proposto consiste nel valutare un'espressione "ben formata" secondo certe regole. Si può procedere sostituendo, a ogni passo, le sotto-espressioni *più interne* con i loro valori, calcolati mediante l'applicazione delle operazioni previste: i passaggi sono illustrati nella figura, in cui in verde è indicato il risultato che si ottiene valutando la espressione in blu collegata con la freccia. Calcolando il valore delle espressioni a mano è facile fare errori di calcolo: per questo motivo è stato attribuito un punteggio parziale anche ad alcune risposte errate, ma probabilmente risultanti da piccoli errori di calcolo.



- Anche questa è informatica -

Il problema proposto richiama l'idea di comporre espressioni, scrivendole l'una nell'altra, ossia **annidandole** in modo appropriato, e ciò ha a che fare con i linguaggi (di programmazione) detti **funzionali**, ma anche con strumenti assai comuni come i fogli di calcolo, in cui le formule sono espressioni in cui appaiono celle di dati o di risultati di altre formule. Questo modello di computazione in cui il risultato di un calcolo deriva unicamente dall'applicazione di una funzione ai suoi operandi è tipico di linguaggi nati nell'ambito dell'intelligenza artificiale come il Lisp, Scheme, Racket, ma oggi è diffuso anche nei linguaggi più comuni come Java e Python. Il modo di scrivere queste espressioni usato nel problema (con le parentesi, il nome della funzione e tutti gli operandi, com'è tipico in Lisp) è detto notazione **prefissa**e, rispetto alla notazione infissa più comune in matematica, ha il vantaggio di non richiedere l'uso di regole di **precedenza** sulle operazioni. Per esempio: (+ 5 (* 3 6)) vale, senza alcuna ambiguità, 23; quando scriviamo (5 + 3 * 6) dobbiamo sapere che la moltiplicazione ha la precedenza sull'addizione e che quindi interpretare l'espressione come 8 * 6 = 48 è sbagliato.

Parole chiave: **linguaggi funzionali, espressioni annidate, precedenze tra operatori**

- Informazioni sul quesito -

Il quesito è stato proposto dal gruppo Bebras Canada (id: 2016-CA-08b)