

1 Esercizio Black Box

Lo scopo di questo esercizio è quello di affinare le vostre abilità aritmetiche. L'esercizio è composto da 2 file: `black_box1.java` e `dataset.txt`. Quello che dovete fare è capire la relazione aritmetica che c'è tra la variabile `x` e il risultato finale guardando la seguente tabella, riportata in `dataset.txt`

1.1 Dataset

X | Output

1 , 0

2 , 0

3 , 1

4 , 2

5 , 0

6 , 4

7 , 3

8 , 2

9 , 1

10 , 0

11 , 10

Una volta che avete capito la relazione aritmetica, per esempio $\text{Output} = x + 10$, dovete inserirla nel file `black_box1.java` nella variabile `value`. A questo punto compilate il file `.java` ed eseguitelo. Se la formula è corretta verrà stampato il messaggio: "La formula inserita è corretta!" altrimenti "la formula inserita NON è corretta".

2 Indovina il numero

Alice incontra Bob e gli dice: "Ho pensato ad un numero tra 0 e 100, proveresti ad indovinarlo?" Bob dice un numero e Alice risponde "Sbagliato! E' un numero più alto." Allora Bob prova con un altro numero e Alice risponde "Sbagliato! E' un numero più basso." Dopo un po' di tentativi falliti Alice dice a Bob "Hai fatto troppi tentativi senza indovinare, non voglio vederti mai più!"

Più tardi, solo nella sua stanza, Bob ripensa a quello che è successo e, preso da un lampo di genio, scrive un programma Java che è in grado di indovinare il numero pensato da Alice nel minor numero di tentativi possibile.

Si scriva il programma di Bob.

2.1 Requisiti

Si implementi un metodo statico `ask()` che preso in input un intero `n`

2.2 Esempio di esecuzione

Pensa ad un numero tra 0 e 100 ...

Alice pensa al numero 33.

Il numero è 50?

Alice risponde:

minore

Il numero è 25?

Alice risponde:

maggiore

Il numero è 37?

Alice risponde:

minore

Il numero è 31?

Alice risponde:

maggiore

Il numero è 34?

Alice risponde:

minore

Il numero è 32?

Alice risponde:

maggiore

Il numero è 33?

Alice risponde:

sì

Ho indovinato!

3 Robot 2D

Un robot mobile segue un percorso specificato da una sequenza di posizioni P_0, P_1, P_2, \dots su un piano 2D dove ogni P_i è specificato da un'ascissa e un'ordinata non negative (ad esempio $P_i = (7, 12)$). Il robot che inizialmente è in base a P_0 , per raggiungere un punto P_i ($i > 0$), si muove lungo la linea retta che congiunge la sua posizione corrente con P_i . Ad esempio, se il percorso fosse P_0, P_1, P_2 , il robot prima percorrerebbe il segmento P_0P_1 , poi il segmento P_1P_2 e infine il segmento P_2P_3 . Il robot dichiara conclusa la sua missione quando una di queste due condizioni si è verificata:

- a. ritorno in base (esempio P_0, P_1, P_2, P_0)
- b. ha visitato 5 punti, senza contare la base ($P_0, P_1, P_2, P_3, P_4, P_5$)

Si scriva un programma simuli l'interfaccia di comando del robot. L'operatore potrà inviare al robot i punti da raggiungere e il robot comunicherà lo stato della missione stampando la distanza totale percorsa.

3.1 Requisiti

1. Nel contare i punti visitati si considerino anche punti già visitati in precedenza dal robot.
2. Si dichiari il numero massimo di punti come una costante del programma.
3. Si implementi un metodo statico `distanza2p` che dati in input due punti P e Q , calcoli la lunghezza del segmento PQ .

3.2 Esempio di esecuzione

Se l'operatore specifica questa sequenza di punti:

```
0      0
12     0
12     50
0      50
```

```
0      60
0      0
```

Il robot invierà questi messaggi:

```
La base è in x0=0 y0=0
Mi trovo in x0=12 y0=0, fino ad ora ho percorso 12 unità di distanza.
Mi trovo in x1=12 y1=50, fino ad ora ho percorso 62 unità di distanza.
Mi trovo in x2=0 y2=50, fino ad ora ho percorso 74 unità di distanza.
Mi trovo in x3=0 y3=60, fino ad ora ho percorso 84 unità di distanza.
Mi trovo in x4=0 y4=0, fino ad ora ho percorso 144 unità di distanza.
Il percorso lungo 144 è terminato: sono ritornato in base.
```

Se l'operatore specifica questa sequenza di punti:

```
0      0
12     0
12     50
0      50
0      60
22     22
100    100
34     76
```

Il robot invierà questi messaggi:

```
La base è in x0=0 y0=0
Mi trovo in x0=12 y0=0. fino ad ora ho percorso 12 unità di distanza.
Mi trovo in x1=12 y1=50. fino ad ora ho percorso 62 unità di distanza.
Mi trovo in x2=0 y2=50. fino ad ora ho percorso 74 unità di distanza.
Mi trovo in x3=0 y3=60. fino ad ora ho percorso 84 unità di distanza.
Mi trovo in x4=22 y4=22. fino ad ora ho percorso 127 unità di distanza.
Mi trovo in x5=100 y5=100. fino ad ora ho percorso 237 unità di distanza.
Il percorso lungo 237 è terminato: ho visitato il numero massimo di punti visitati.
```

4 Codice Crittografico Cesare

Il codice cesare è un sistema usato per proteggere dei messaggi scambiati tra due persone. Il codice funziona trasformando una stringa nella sua corrispondente cifrata di k posizioni.

a b c d e f g h i l m n s t u v z

Per esempio $k = 2$ casa ==> ecuc

per ritornare al messaggio originale basta applicare la funzione inversa.

Fare due programmi cifra e decifra. Il primo programma, cifra, prende in input la stringa da proteggere insieme un numero rappresentante lo shift e stampa

in output la stringa cifrata. Il secondo programma prende in input la stringa cifrata e lo shift utilizzato per la cifratura e restituisce la stringa originale.

5 Esercizio Black Box

Lo scopo di questo esercizio è quello di affinare le vostre abilità aritmetiche. L'esercizio è composto da 2 file: `black_box2.java` e `dataset.txt`. Quello che dovete fare è capire la relazione aritmetica che c'è tra la variabile `x` e il risultato finale guardando la seguente tabella, riportata in `dataset.txt`

5.1 Dataset

X | Output

1 , 2

2 , 3

3 , 1

4 , 2

5 , 3

6 , 1

7 , 2

8 , 3

9 , 1

10 , 2

11 , 3

12 , 1

13 , 2

14 , 3

15 , 1

16 , 2

17 , 3

18 , 1

19 , 2

20 , 3

21 , 1

22 , 2

23 , 3

Una volta che avete capito la relazione aritmetica, per esempio $\text{Output} = x + 10$, dovete inserirla nel file `black_box1.java` nella variabile *value*. A questo punto compilate il file `.java` ed eseguitelo. Se la formula è corretta verrà stampato il messaggio: “La formula inserita è corretta!” altrimenti “la formula inserita NON è corretta”.