

Esercitazione 7

Array

Programmazione e Analisi di Dati
Mod. A – Programmazione Java

Esercizio 1. Modificare il programma `RipetoDieciNumeri2` visto a lezione in modo che stampi i numeri inseriti in ordine inverso.

Esempio di esecuzione:

```
Inserisci 10 numeri interi:  
7  
6  
12  
667  
-33  
0  
13  
-34  
4  
100  
Risultato:  
100  
4  
-34  
13  
0  
-33  
667  
12  
6  
7
```

Esercizio 2. Modificare l'esercizio precedente aggiungendo la richiesta all'utente di quanti numeri inserire.

Esempio di esecuzione:

```
Quanti numeri vuoi inserire?  
5  
Inserisci 5 numeri interi:  
12  
667  
13  
-34  
100  
Risultato:  
100  
-34  
13  
667  
12
```

Esercizio 3. Scrivere il programma `MinimoDiArray` che chiede all'utente di inserire 10 numeri, li memorizza in un array e poi stampa valore e posizione dell'elemento minimo dell'array. Se il valore minimo è presente più di una volta, stampa la posizione della prima occorrenza di tale valore nell'array.

Esempio di esecuzione:

```
Inserisci 10 numeri interi:
7
6
12
667
-33
0
13
-34
4
-34
Il valore minimo e' -34 in posizione 7
```

Esercizio 4. Modificare l'esercizio precedente in modo che utilizzi stringhe (singole parole) invece che numeri e stampi valore e posizione della prima stringa secondo l'ordine lessicografico.

Esercizio 5. Scrivere il programma `Cerca` che crea e inizializza un array di `int` contenete 10 valori a piacere (usando il comando di inizializzazione che prevede la lista dei valori tra parentesi graffe), chiede all'utente di inserire un numero e stampa `Presente` se il numero è presente nell'array, oppure `Assente` se il numero non è presente. Provare a risolvere questo esercizio in due modi: tramite un `for-each` e tramite un `while`. Nel caso del `while` il ciclo deve interrompersi non appena l'elemento (se presente) viene trovato.

Esempi di esecuzione (supponendo che i numeri inseriti siano 18, 9, 20, 211, 22, 0, -12, -83, 29, 5):

```
Inserire un numero:
100
Assente!
```

```
Inserire un numero:
20
Presente!
```

Esercizio 6. Scrivere il programma `UgualiConsecutivi` che chiede all'utente di inserire 10 numeri, li memorizza in un array e stampa valori e posizioni di tutti gli elementi che sono preceduti dallo stesso elemento (Ricordare che le posizioni iniziano da 0). Se non ci sono coppie di valori consecutivi uguali il programma non stampa nulla.

Esempio di esecuzione:

```
Inserisci 10 numeri interi:
8
12
34
8
8
9
9
9
90
4
Il valore 8 in posizione 4 e' uguale al precedente
Il valore 9 in posizione 6 e' uguale al precedente
Il valore 9 in posizione 7 e' uguale al precedente
```

Esercizio 7. Scrivere un programma `Temperature` che crea e inizializza un array di `double` contenete 12 valori a piacere che rappresentano temperature atmosferiche (usando il comando di inizializzazione che prevede la lista dei valori tra parentesi graffe). Il programma calcola la media di tali temperature e, per ogni valore di temperatura nell'array, stampa un messaggio che dice se essa è maggiore (o uguale) oppure inferiore alla media.

Esempio di esecuzione (supponendo che le temperature inserite siano 18.0, 19.5, 20.0, 21.0, 22.3, 24.0, 23.6, 22.8, 22.9, 20.5, 19.2, 18.7):

```
18.0 inferiore alla media
19.5 inferiore alla media
20.0 inferiore alla media
21.0 inferiore alla media
22.3 superiore o uguale alla media
24.0 superiore o uguale alla media
23.6 superiore o uguale alla media
22.8 superiore o uguale alla media
22.9 superiore o uguale alla media
20.5 inferiore alla media
19.2 inferiore alla media
18.7 inferiore alla media
```

Esercizio 8. Scrivere un programma `ParoleMisteriose` che crea e inizializza un array di stringhe contenete 5 parole a piacere (usando il comando di inizializzazione che prevede la lista dei valori tra parentesi graffe) e chiede all'utente di indovinare una tra le 5 parole entro 3 tentativi.

Esempi di esecuzione (supponendo che le parole da indovinare siano `cane`, `gatto`, `topo`, `criceto` e `pesce`):

```
Inserire una parola:
rana
Sbagliato!
Inserire una parola:
gatto
Indovinato!
```

```
Inserire una parola:
mucca
Sbagliato!
Inserire una parola:
ragno
Sbagliato!
Inserire una parola:
tafano
Sbagliato!
Tentativi esauriti
```

Esercizio 9. Modificare l'esercizio precedente in modo che se l'utente indovina una parola il programma stampi l'elenco delle altre parole misteriose, mentre se l'utente sbaglia stampi tutte e 5 le parole misteriose

Esempi di esecuzione (supponendo che le parole da indovinare siano `cane`, `gatto`, `topo`, `criceto` e `pesce`):

```
Inserire una parola:
rana
Sbagliato!
Inserire una parola:
gatto
Indovinato!
Le altre parole misteriose sono: cane, topo, criceto e pesce
```

```
Inserire una parola:
mucca
Sbagliato!
Inserire una parola:
ragno
Sbagliato!
Inserire una parola:
tafano
Sbagliato!
Tentativi esauriti
Le parole misteriose erano: cane, gatto, topo, criceto e pesce
```

Esercizio 10 (ARTICOLATO). Scrivere un programma modulare `Memo` che, utilizzando un array, legge 10 stringhe e poi mette a disposizione un menu con le seguenti opzioni:

```
[S] Stampa stringhe dalla posizione ... alla ...  
[M] Modifica stringa in posizione ...  
[T] Trova posizione della stringa ...  
[E] Esci
```

Il menu viene proposto ripetutamente all'utente, fino a quando viene selezionata l'uscita dal programma (opzione E). Il significato delle operazioni è il seguente:

- se l'utente sceglie S, il programma legge due posizioni n e m e stampa tutti gli elementi dell'array il cui indice è compreso tra n e m ;
- se l'utente sceglie M, il programma legge una posizione n e una stringa s che assegna alla cella n dell'array;
- se l'utente sceglie T, il programma legge una stringa s e stampa la prima posizione dell'array che la contiene, o un messaggio di errore se nessuna cella la contiene;
- se l'utente sceglie E, il programma stampa il contenuto dell'array e termina.

Ognuna delle operazioni (tranne [E]) deve essere realizzata da un metodo diverso che realizza quanto indicato.