

```

1  /*
2  * -- M.G. Maffucci --
3  * 11.
4  * 11.1. Si ha un elenco di articoli di magazzino formato da descrizione,
5  * giacenza, prezzo acquisto, prezzo vendita IVA esclusa.
6  *
7  * 11.2. Determinare la valorizzazione di magazzino
8  *
9  * 11.3. e stampare un listino alfabetico riportante descrizione, prezzo di
10 * vendita senza IVA e con IVA 20%.
11 *
12 * NOTA: l'esercizio richiede un calcolo aritmetico iniziale e un algoritmo di
13 * ordinamento degli elementi di un vettore.
14 */
15
16 #include <stdio.h>
17 #include <stdlib.h>
18 #include <string.h>
19
20 #define NMAX 100          //numero massimo di prodotti
21 #define LMAX 20          //lunghezza massima della descrizione del prodotto
22
23 int main()
24 {
25     //input
26     char descrizioni[NMAX][LMAX]; //descrizioni dei prodotti
27     float giacenze[NMAX];         //giacenze dei prodotti
28     float pzoa[NMAX];             //prezzi di acquisto dei prodotti
29     float pzov[NMAX];             //prezzi di vendita dei prodotti IVA
30                                   // esclusa
31     int n;                        //numero di prodotti da inserire
32     //lavoro
33     int i;                        //indice dei vettori
34     int j;                        //indice dei vettori usato per l'ordinamento
35     char descSup[LMAX];           //variabile di supporto utilizzata per
36                                   // l'ordinamento del vettore descrizioni[]
37     float pzovSup;               //variabile di supporto utilizzata per
38                                   // l'ordinamento del vettore pzov[]
39     //output
40     float valMag;                //valore del magazzino
41     /*
42     * 11.1. Si ha un elenco di articoli di magazzino formato da descrizione,
43     * giacenza, prezzo acquisto, prezzo vendita IVA esclusa.
44     * Inserisco il numero di articoli da caricare, controllando la validita'
45     * dell'input.
46     */
47     do{
48         printf("Inserisci il numero di articoli da caricare (1-100): ");
49         scanf("%d", &n);
50     } while( (n < 1) || (n > NMAX));
51
52     /*
53     * Ciclo di caricamento dei quattro vettori.
54     * Il caricamento viene fatto parallelamente, in un unico ciclo.
55     */
56     for(i = 0; i < n; i++){
57         while(getchar() != '\n');
58         printf("Inserisci la descrizione dell'articolo: ");
59         fgets(descrizioni[i], LMAX, stdin);
60         descrizioni[i][strlen(descrizioni[i]) - 1] = '\0';
61         printf("Inserisci la giacenza dell'articolo: ");
62         scanf("%f", &giacenze[i]);
63         printf("Inserisci il prezzo di acquisto dell'articolo: ");
64         scanf("%f", &pzoa[i]);
65         printf("Inserisci il prezzo di vendita dell'articolo: ");
66         scanf("%f", &pzov[i]);

```

```

67     }
68     /*
69     * 11.2. Determinare la valorizzazione di magazzino
70     */
71     for( i = 0, valMag = 0; i < n; i++){
72         valMag = valMag + giacenze[i] * pzoa[i];
73     }
74     printf("\nValorizzazione del magazzino: %.2f", valMag);
75     /*
76     * 11.3. e stampare un listino alfabetico riportante descrizione, prezzo di
77     * vendita senza IVA e con IVA 20%.
78     *
79     * Ordiniamo i vettori.
80     */
81     for(i = 0; i < (n - 1); i++){
82         for(j = (i + 1); j < n; j++){
83             if(strcmp(descrizioni[i], descrizioni[j]) > 0){
84                 /*
85                 * Ordino la descrizione.
86                 */
87                 strcpy(descSup, descrizioni[i]);
88                 strcpy(descrizioni[i], descrizioni[j]);
89                 strcpy(descrizioni[j], descSup);
90                 /*
91                 * Ordino la giacenza.
92                 * Questa operazione si puo' evitare perche' una volta conclusa
93                 * l'esecuzione del programma tutti i dati nei vettori saranno
94                 * persi, ma e' bene ricordare che se la memorizzazione dei dati
95                 * fosse permanente, dovremmo obbligatoriamente effettuare
96                 * l'ordinamento anche della giacenza, pena l'inconsistenza dei
97                 * dati.
98                 */
99
100                /*
101                * Ordino il prezzo di acquisto.
102                * Questa operazione si puo' evitare perche' una volta conclusa
103                * l'esecuzione del programma tutti i dati nei vettori saranno
104                * persi, ma e' bene ricordare che se la memorizzazione dei dati
105                * fosse permanente, dovremmo obbligatoriamente effettuare
106                * l'ordinamento anche del prezzo di acquisto, pena
107                * l'inconsistenza dei dati.
108                */
109
110                /*
111                * Ordino il prezzo di vendita senza IVA.
112                */
113                pzovSup = pzov[i];
114                pzov[i] = pzov[j];
115                pzov[j] = pzovSup;
116            }
117        }
118    }
119    printf("\nListino vendita\n");
120    printf("-----\n");
121    printf("|Descrizione          |Prezzo senza IVA|Prezzo con IVA|\n");
122    printf("-----\n");
123    for(i = 0; i < n; i++){
124        printf("|%-20.20s|          %8.2f|          %8.2f|\n", descrizioni[i], pzov[i]
125            , (pzov[i]*1.2));
126    }
127    printf("-----\n");
128
129    return 0;
130 }

```