

# Esercizi sui vettori in C

Stefano Cherubin\*

30/10/2015

[**Informatica A**] Esercitazione #6

corso per Ing. Gestionale a.a. 2015/16

---

\*<nome.cognome>@polimi.it

## Indice

<b>1</b>	<b>Analisi del fatturato</b>	<b>3</b>
1.1	Approccio alla soluzione . . . . .	3
1.2	Posizione o valore . . . . .	3
1.3	Soluzione C - Store and scan . . . . .	4
1.4	Variante: confronto con l'anno passato . . . . .	5
<b>2</b>	<b>Merge</b>	<b>7</b>
2.1	Soluzione C . . . . .	7
<b>3</b>	<b>Ordinamento di vettori</b>	<b>9</b>
3.1	Approccio alla soluzione . . . . .	9
3.2	Soluzione C - Selection Sort . . . . .	9

# 1 Analisi del fatturato

Sono arrivati dal reparto vendite i dati relativi all'andamento del volume di mercato dell'ultimo anno. I dati rappresentano le quantità vendute per ogni mese dell'anno. Si calcoli il mese con il massimo volume di vendita e la più lunga serie crescente.

## 1.1 Approccio alla soluzione

Il problema richiede la ricerca del valore massimo e della più lunga sequenza di valori crescenti.

È possibile

Non sapendo le unità di misura del volume, si può fare una assunzione esplicita che si tratti di unità vendute e quindi utilizzare il tipo **int** oppure utilizzare il tipo **float** senza imporre vincoli o assunzioni sulle unità di misura dei dati in ingresso. In entrambi i casi, trattandosi di quantità nette vendute, si dovrà verificare che i dati in ingresso siano positivi o nulli.

## 1.2 Posizione o valore

Quando si eseguono operazioni di ricerca (di un elemento, del massimo, del minimo, di un intervallo, ...) all'interno di un vettore si può scegliere se lavorare con il valore o con la posizione. Ad esempio durante la ricerca del massimo, si può memorizzare il valore massimo e aggiornarlo ogni volta.

```
for (i = 0; i < N; ++i) {  
    if (v[i] > max) {  
        max = v[i];  
    }  
}
```

Un altro approccio è quello di aggiornare la posizione del valore massimo; utilizzando la posizione, è sempre possibile tornare al valore massimo e consente di occupare in memoria solamente una variabile per l'indice relativo alla posizione dell'elemento di interesse. Inoltre risulta più semplice effettuare ordinamenti più complessi (ordinamento su più livelli, ordinamento di elementi complessi, ...).

```
for (i = 0; i < N; ++i) {  
    if (v[i] > v[pmax]) {  
        pmax = i;  
    }  
}
```

### 1.3 Soluzione C - Store and scan

Listato 1: Analisi fatturato

```
1 #include <stdio.h>
2 #define L 12
3 int main () {
4     float volume[L];
5     int i, p1, p2, current_p1, current_p2, pmax;
6     for (i = 0; i < L; ++i) {
7         printf("\nVolume del mese %d: ", i);
8         scanf("%f", &volume[i]);
9         if (volume[i] < 0) {
10             printf("\nVolumi negativi non ammessi");
11             --i;
12         }
13     }
14     pmax = 0;
15     p1 = 0;
16     p2 = 0;
17     current_p1 = 0;
18     current_p2 = 0;
19     for (i = 1; i < L; ++i) {
20         if (volume[pamx] < volume[i]) {
21             pmax = i;
22         }
23         if (volume[i] > volume[i - 1]) {
24             current_p2 = i;
25             if (current_p1 == 0) {
26                 current_p1 = i - 1;
27             }
28             if ((current_p2 - current_p1) > (p2 - p1)
29                 ) {
30                 p2 = current_p2;
31                 p1 = current_p1;
32             }
33         } else {
34             current_p1 = 0;
35             current_p2 = 0;
36         }
37     }
38     printf("\nIl massimo si è avuto al mese %d
39         con %f", pmax, volume[pmax]);
```

```

39     printf("\nLa più lunga sequenza positiva è
        dal mese %d al mese %d.", p1, p2);
40     return 0;
41 }

```

#### 1.4 Variante: confronto con l'anno passato

Si vuole confrontare i dati di quest'anno con i dati dell'anno scorso. Si acquisiscano due sequenze di volumi. Si assuma che questi volumi rappresentino unità di prodotto vendute e siano interi positivi (o nulli). Si determini in quale mese si è avuto l'incremento massimo rispetto al corrispettivo del precedente anno e si identifichi la più lunga sequenza di incrementi positivi.

Listato 2: Analisi fatturato: confronto con l'anno passato

```

1  #include <stdio.h>
2  #define L 12
3  int main () {
4      float inc[L];
5      int volume[L];
6      int old[L];
7      int i, p1, p2, current_p1, current_p2, pmax;
8      for (i = 0; i < L; ++i) {
9          printf("\nVolume del mese %d per l'anno
                corrente: ", i);
10         scanf("%d", &volume[i]);
11         if (volume[i] < 0) {
12             printf("\nVolumi negativi non ammessi");
13             --i;
14         }
15     }
16     for (i = 0; i < L; ++i) {
17         printf("\nVolume del mese %d per l'anno
                passato: ", i);
18         scanf("%d", &old[i]);
19         if (old[i] < 0) {
20             printf("\nVolumi negativi non ammessi");
21             --i;
22         }
23     }
24     for (i = 0; i < L; ++i) {
25         if (old[i] != 0) { /* evita divisione per 0
                */
26             inc[i] = (float) (volume[i] - old[i]) /
                old[i];

```

```

27     } else {
28         inc[i] = 9999.9; /* valore di comodo */
29     }
30 }
31 pmax = 0;
32 p1 = -1;
33 p2 = -2;
34 current_p1 = -1;
35 current_p2 = -1;
36 for (i = 0; i < L; ++i) {
37     if (inc[pmax] < inc[i]) {
38         pmax = i;
39     }
40     if (inc[i] > 0.0) {
41         current_p2 = i;
42         if (current_p1 == -1) {
43             current_p1 = i;
44         }
45         if ((current_p2 - current_p1) > (p2 - p1)
46             ) {
47             p2 = current_p2;
48             p1 = current_p1;
49         }
50     } else {
51         current_p1 = -1;
52         current_p2 = -2;
53     }
54 }
55 printf("\nIl massimo si è avuto al mese %d
56       con %f", pmax, inc[pmax]);
57 if (p1 != -1) {
58     printf("\nLa più lunga sequenza positiva è
59           dal mese %d al mese %d.", p1, p2);
60 } else {
61     printf("\nNon esiste una sequenza positiva
62           rispetto al precedente anno");
63 }

```

## 2 Merge

Scrivere un programma in linguaggio C che lette dallo standard input due sequenze vettoriali ordinate di interi  $V1[n]$ ,  $V2[m]$  ne crei una terza  $V3[n+m]$  anch'essa ordinata, che contenga tutti gli elementi di  $V1$  e di  $V2$ .

### 2.1 Soluzione C

Sfruttando il fatto che le sequenze  $V1$  e  $V2$  sono già ordinate, posso scandirle progressivamente entrambe e per ogni elemento da inserire in  $V3$  scegliere l'elemento minore tra le due sorgenti. Quando una delle due sorgenti termina, esaurisco gli elementi residui copiandoli in coda al risultato  $V3$ .

Listato 3: Merge

```
1 #include <stdio.h>
2 #define MAX_LEN 50
3 int main( ) {
4     int V1[MAX_LEN], V2[MAX_LEN];
5     int V3[2*MAX_LEN];
6     int n, m;
7     int i, j, k;
8     /* acquisizione dei due vettori */
9     do {
10         printf("\nInserire il # di elementi del
            vettore V1 (max %d), n = ", MAX_LEN);
11         scanf("%d",&n);
12     } while ((n <= 0) || (n > MAX_LEN));
13     printf("\nInserire gli elementi ordinati del
            vettore V1:\n");
14     for (i = 0; i < n; ++i) {
15         scanf("%d", &V1[i]);
16     }
17     do {
18         printf("\nInserire il # di elementi del
            vettore V2 (max %d), m = ", MAX_LEN);
19         scanf("%d",&m);
20     } while ((m <= 0) || (m > MAX_LEN));
21     printf("\nInserire gli elementi ordinati del
            vettore V2:\n");
22     for (j = 0; j < m; ++j)
23         scanf("%d", &V2[j]);
24     }
25     /* fusione delle due sequenze */
26     i = 0;
27     j = 0;
```

```

28     k = 0;
29     while ((i < n) && (j < m)) {
30         if (V1[i] < V2[j]) {
31             V3[k] = V1[i];
32             i++;
33         } else {
34             V3[k] = V2[j];
35             j++;
36         }
37         k++;
38     }
39     while (i < n) {
40         V3[k] = V1[i];
41         i++;
42         k++;
43     }
44     while (j < m) {
45         V3[k] = V2[j];
46         j++;
47         k++;
48     }
49     printf("\nIl risultato della fusione dei due
        vettori e':\n");
50     for (k = 0; k < n + m; ++k) {
51         printf("%d ", V3[k]);
52     }
53     return 0;
54 }

```



### 3 Ordinamento di vettori

Data una sequenza vettoriale  $V[n]$  permutare i suoi elementi in modo che risulti ordinata in senso non decrescente.

Esempio:  $v[0] \leq v[1] \leq v[2] \leq \dots \leq v[n-1]$ .

#### 3.1 Approccio alla soluzione

Esistono diversi algoritmi di ordinamento di vettori. Di seguito ne viene presentato uno, tra i più semplici: l'ordinamento per selezione.

L'idea è quella di trovare il minimo fra gli elementi nella sequenza e scambiarlo con l'elemento al primo posto. Trovare il minimo fra gli elementi tra il secondo e l'ultimo posto, e scambiarlo con il secondo e così via. Il metodo è chiamato *per selezione* perché si basa sulla ripetizione della selezione dell'elemento minore, tra quelli rimasti da ordinare.

#### 3.2 Soluzione C - Selection Sort

Listato 4: Ordinamento per selezione

```
1 #include <stdio.h>
2 #define MAX_LEN 100
3 int main ( ) {
4     int V[MAX_LEN];
5     int n, min;
6     int i, j, jmin;
7     do {
8         printf("\nInserire il # di elementi del
9             vettore V, n = ");
10        scanf("%d", &n);
11    } while ((n <= 0) || (n > MAX_LEN));
12    printf("\nInserire gli elementi del vettore V
13        :\n");
14    for (i = 0; i < n; ++i) {
15        scanf("%d", &V[i]);
16    }
17    for (i = 0; i < n; i++) {
18        jmin = i;
19        min = V[jmin];
20        for (j = i + 1; j < n; j++) {
21            if (V[j] < min) {
22                jmin = j;
23                min = V[j];
24            }
25        }
26    }
```

```

24      /* min ora contiene il minimo parziale, cioè
        il minimo calcolato dalla posizione i alla
        fine del vettore */
25      V[jmin] = V[i];
26      V[i] = min; /* scambio V[i] e V[jmin] */
27  }
28  printf("\n il vettore ordinato e':\n ");
29  for (i = 0; i < n; i++) {
30      printf("%d ", V[i]);
31  }
32  return 0;
33  }

```

## **Licenza e crediti**

### **Crediti**

Quest'opera contiene elementi tratti da materiale di Gerardo Pelosi redatto per il corso di Fondamenti di Informatica per Ingegneria dell'Automazione a.a. 2013/14.

### **Licenza beerware<sup>1</sup>**

Quest'opera è stata redatta da Stefano Cherubin. Mantenendo questa nota, puoi fare quello che vuoi con quest'opera. Se ci dovessimo incontrare e tu ritenessi che quest'opera lo valga, in cambio puoi offrirmi una birra.

---

<sup>1</sup><http://people.freebsd.org/~phk/>