

Corso di Algoritmi e Strutture Dati (IN110) – Prof. Marco Liverani – a.a. 2024/2025

## Esame scritto del 12 Settembre 2025 (Appello X)

Si richiede di risolvere entrambi gli esercizi riportando una codifica in linguaggio C completa dei due programmi. Nel caso in cui non si riesca a completare entrambi gli esercizi si suggerisce di riportare almeno la codifica in C delle funzioni principali o una loro pseudo-codifica. È possibile consultare libri e appunti personali, ma non scambiare libri o appunti con altri studenti. I compiti che presenteranno evidenti ed anomale «similitudini» saranno annullati. La prova scritta ha una durata di tre ore, durante le quali non è consentito allontanarsi dall'aula, se non dopo aver consegnato il compito.

Deve essere consegnata solo la «bella copia» del compito scritto; su ciascun foglio deve essere riportato il **nome**, il **cognome** e il **numero di matricola** dello studente.

### Esercizio n. 1

Letta in input una sequenza di  $n$  numeri interi, memorizzarla in una lista  $L_1$ . Generare una sequenza di  $m$  numeri casuali positivi minori di 100 memorizzandoli in una seconda lista  $L_2$ . Stampare entrambe le liste. Per ogni nodo della lista  $L_2$  memorizzare in uno dei suoi campi il numero di occorrenze dello stesso elemento nella lista  $L_1$ . Se un elemento di  $L_2$  non è presente in  $L_1$ , allora deve essere eliminato da  $L_2$ . Stampare in output la lista  $L_2$  al termine dell'elaborazione.

**Esempio** Sia  $n = 10$  e sia  $L_1 = \langle 26, 12, 67, 58, 12, 63, 67, 12 \rangle$ .

Sia  $L_2 = \langle (62, 0), (95, 0), (58, 0), (24, 0), (12, 0), (67, 0), (15, 0), (24, 0), (63, 0), (17, 0), (12, 0) \rangle$  la lista di numeri casuali.

Al termine dell'elaborazione la lista  $L_2$  è la seguente:  $L_2 = \langle (58, 1), (12, 3), (67, 2), (63, 1), (12, 3) \rangle$ .

### Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4
5 struct nodo1 {
6     int info;
7     struct nodo1 *next;
8 };
9
10 struct nodo2 {
11     int info, n;
12     struct nodo2 *next;
13 };
14
15 struct nodo1 *leggi_lista(void) {
16     int n, x, i;
17     struct nodo1 *p, *primo;
18     printf("Numero di elementi: ");
19     scanf("%d", &n);
20     primo = NULL;
21     printf("Inserisci %d numeri minori di 100:\n", n);
22     for (i=0; i<n; i++) {
23         scanf("%d", &x);
24         p = malloc(sizeof(struct nodo1));

```

```

25     p->info = x;
26     p->next = primo;
27     primo = p;
28 }
29 return primo;
30 }
31
32 struct nodo2 *genera_lista(void) {
33     int m, x, i;
34     struct nodo2 *p, *primo = NULL;
35     srand((unsigned)time(NULL));
36     m = rand() % 100;
37     for (i=0; i<m; i++) {
38         x = rand() % 100;
39         p = malloc(sizeof(struct nodo2));
40         p->info = x;
41         p->n = 0;
42         p->next = primo;
43         primo = p;
44     }
45     return primo;
46 }
47
48 void stampa_lista(struct nodo2 *p) {
49     while (p != NULL) {
50         printf("(%d, %d) -->", p->info, p->n);
51         p = p->next;
52     }
53     printf("NULL\n\n");
54     return;
55 }
56
57 int main(void) {
58     struct nodo1 *primo1, *p;
59     struct nodo2 *primo2, *q, *q1;
60     primo1 = leggi_lista();
61     primo2 = genera_lista();
62     stampa_lista(primo2);
63     q = NULL;
64     q1 = primo2;
65     while (q1 != NULL) {
66         p = primo1;
67         while (p != NULL) {
68             if (q1->info == p->info)
69                 q1->n += 1;
70             p = p->next;
71         }
72         if (q1->n == 0) {
73             if (q == NULL) {
74                 primo2 = q1->next;
75                 free(q1);
76                 q1 = primo2;
77             } else {

```

```

78     q->next = q1->next;
79     free(q1);
80     q1 = q->next;
81 }
82 } else {
83     q = q1;
84     q1 = q1->next;
85 }
86 }
87 stampa_lista(primo2);
88 return 0;
89 }

```

## Esercizio n. 2

Letto in input un intero  $n > 0$ , generare un vettore  $A$  di  $n$  numeri interi casuali compresi nell'insieme  $\{-17, -16, -15, \dots, 7, 8, 9\}$ . Costruire un secondo vettore  $B$  con  $n - 1$  elementi tale che

$$B_i = \max\{A_0, \dots, A_i\} - \min\{A_{i+1}, \dots, A_{n-1}\}$$

**Esempio** Sia  $n = 6$  e  $A = (4, 6, 2, -13, 8, 2)$ ; allora  $B = (17, 19, 19, 4, 6)$ .

## Soluzione

```

1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <time.h>
4  #define MAX 30
5
6  void stampa_array(int X[], int n) {
7      for (int i=0; i<n; i++)
8          printf("%d ", X[i]);
9      printf("\n");
10     return;
11 }
12
13 int genera_array(int X[]) {
14     int i, n;
15     srand((unsigned)time(NULL));
16     printf("Numero di elementi: ");
17     scanf("%d", &n);
18     for (i=0; i<n; i++)
19         X[i] = rand() % 27 - 17;
20     return n;
21 }
22
23 int min(int X[], int a, int b) {
24     int m;
25     m = X[a];
26     while (a<b) {
27         a++;
28         if (X[a] < m)
29             m = X[a];

```

```

30 }
31 return m;
32 }
33
34 int max(int X[], int a, int b) {
35     int m;
36     m = X[a];
37     while (a<b) {
38         a++;
39         if (X[a] > m)
40             m = X[a];
41     }
42     return m;
43 }
44
45 void costruisci(int A[], int n, int B[]) {
46     for (int i=0; i<n-1; i++)
47         B[i] = max(A, 0, i) - min(A, i+1, n-1);
48     return;
49 }
50
51 int main(void) {
52     int A[MAX], B[MAX], n;
53     n = genera_array(A);
54     stampa_array(A, n);
55     costruisci (A, n, B);
56     stampa_array(B, n-1);
57     return 0;
58 }

```