

Corso di Algoritmi e Strutture Dati (IN110) – Prof. Marco Liverani – a.a. 2025/2026

Esame scritto del 11 Settembre 2026 (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 o utilizzare strumenti digitali (tablet, smartphone, computer, ecc.). 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

In una matrice con n colonne e 3 righe sono raccolti i risultati di un campionato a k squadre al termine della n -esima partita: su ciascuna colonna è riportato il risultato di una partita, indicando nella prima riga il codice numerico progressivo della squadra (da 1 al numero di squadre) che ha giocato in casa, nella seconda riga il codice della squadra che ha giocato fuori casa e nella terza il risultato (0 = pareggio, 1 = vittoria della squadra di casa, 2 = vittoria della squadra ospite).

Tenendo conto che una vittoria vale 2 punti, un pareggio 1 punto e una sconfitta 0 punti, calcolare il codice di una delle squadre al vertice della classifica, dopo aver acquisito in input il numero $n > 0$ di partite, il numero $k > 0$ di squadre e la matrice dei risultati.

Esempio Sia $n = 9$, $k = 6$ e si consideri la seguente matrice dei risultati:

$$A = \begin{pmatrix} 1 & 3 & 5 & 1 & 2 & 4 & 1 & 3 & 2 \\ 2 & 4 & 6 & 3 & 5 & 6 & 5 & 6 & 4 \\ 0 & 1 & 2 & 1 & 1 & 2 & 0 & 0 & 2 \end{pmatrix}$$

La squadra al vertice della classifica è la n.6 con 5 punti (2 vittorie, 1 pareggio, 0 sconfitte).

Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 50
5
6 void leggiMatrice(int A[3][MAX], int n) {
7     int i;
8     printf("Inserisci i codici delle squadre e l'esito dell'incontro:\n");
9     for (i=0; i<n; i++)
10         scanf("%d %d %d", &A[0][i], &A[1][i], &A[2][i]);
11     return;
12 }
13
14 void calcolaVincitore(int A[3][MAX], int n) {
15     int punti[MAX], i, max;

```

```

16 max = A[0][0];
17 for(i=0; i<MAX; i++)
18     punti[i] = 0;
19 for (i=0; i<n; i++) {
20     if (A[2][i] == 0) {
21         punti[A[0][i]] += 1;
22         punti[A[1][i]] += 1;
23     } else if (A[2][i] == 1) {
24         punti[A[0][i]] += 2;
25     } else {
26         punti[A[1][i]] += 2;
27     }
28     if (punti[A[0][i]] > punti[max])
29         max = A[0][i];
30     if (punti[A[1][i]] > punti[max]) {
31         max = A[1][i];
32     }
33 }
34 printf("\nLa squadra %d e' in testa con %d punti.\n", max, punti[max]);
35 return;
36 }
37
38 void stampaMatrice(int A[3][MAX], int n) {
39     int i, j;
40     printf("\n");
41     for (i=0; i<3; i++) {
42         for (j=0; j<n; j++)
43             printf("| %2d ", A[i][j]);
44         printf("|\n");
45     }
46     return;
47 }
48
49 int main(void) {
50     int n, k, A[3][MAX];
51     printf("Numero di partite disputate: ");
52     scanf("%d", &n);
53     printf("Numero di squadre: ");
54     scanf("%d", &k);
55     leggiMatrice(A, n);
56     stampaMatrice(A, n);
57     calcolaVincitore(A, n);
58     return 0;
59 }

```

Esercizio n. 2

Letti in input due interi $n, k > 0$ generare una sequenza di n interi casuali compresi tra $-k$ e $2k$, estremi inclusi, e memorizzarli in una lista. Stampare la lista. Modificare la lista inserendo un elemento in mezzo ad ogni coppia di elementi preesistenti nella lista originale, il cui valore sia la media aritmetica degli altri due. Stampare la lista modificata in questo modo.

Esempio Siano $n = 8$ e $k = 5$. Se la lista di numeri casuali originale è la seguente:

$\boxed{4} \rightarrow \boxed{-3} \rightarrow \boxed{9} \rightarrow \boxed{7} \rightarrow \boxed{0} \rightarrow \boxed{-1} \rightarrow \boxed{6} \rightarrow \boxed{10}$

la lista prodotta dal programma è la seguente:

$\boxed{4} \rightarrow \boxed{0,5} \rightarrow \boxed{-3} \rightarrow \boxed{3} \rightarrow \boxed{9} \rightarrow \boxed{8} \rightarrow \boxed{7} \rightarrow \boxed{3,5} \rightarrow \boxed{0} \rightarrow \boxed{-0,5} \rightarrow \boxed{-1} \rightarrow \boxed{2,5} \rightarrow \boxed{6} \rightarrow \boxed{8} \rightarrow \boxed{10}$

Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4
5 #define MAX 100
6
7 struct nodo {
8     float info;
9     struct nodo *next;
10 };
11
12 void stampaLista(struct nodo *p) {
13     while (p != NULL) {
14         printf("%.2f --> ", p->info);
15         p = p->next;
16     }
17     printf("NULL\n");
18     return;
19 }
20
21 struct nodo *costruisciLista() {
22     int n, i, k;
23     struct nodo *p, *primo=NULL;
24     printf("Numero di elementi: ");
25     scanf("%d", &n);
26     printf("Valore soglia: ");
27     scanf("%d", &k);
28     srand((unsigned)time(NULL));
29     for (i=0; i<n; i++) {
30         p = malloc(sizeof(struct nodo));
31         p->next = primo;
32         primo = p;
33         p->info = rand() % (3*k+1) - k;
34     }
35     return primo;
```

```

36 }
37
38 void modificaLista(struct nodo *p) {
39     struct nodo *q;
40     while (p->next != NULL) {
41         q = malloc(sizeof(struct nodo));
42         q->info = (p->info + p->next->info)/2.0;
43         q->next = p->next;
44         p->next = q;
45         p = q->next;
46     }
47     return;
48 }
49
50 int main(void) {
51     struct nodo *p;
52     p = costruisciLista();
53     stampaLista(p);
54     modificaLista(p);
55     stampaLista(p);
56     return 0;
57 }

```