

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

## Esame scritto del 10 Luglio 2026 (Appello D)

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

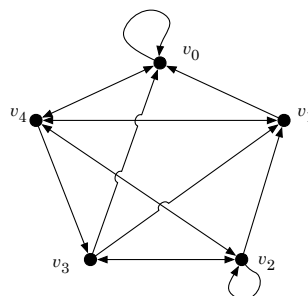
Letto in input un intero  $n > 0$ , costruire una matrice quadrata  $A$  di ordine  $n$  composta da numeri casuali in  $\{-1, 0, 1\}$ . Costruire le liste di adiacenza del grafo orientato  $G = (V, E)$ , tale che se

$$A_{i,j} = \begin{cases} -1 & \implies (v_j, v_i) \in E(G) \\ 1 & \implies (v_i, v_j) \in E(G) \end{cases}$$

avendo cura di non creare spigoli doppi (possono invece essere presenti dei cappi). Visualizzare in output la matrice  $A$  e le liste di adiacenza di  $G$ .

**Esempio** Sia  $n = 5$  e si consideri la matrice  $A$  in figura. Il grafo corrispondente è rappresentato sulla destra.

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & -1 \\ -1 & -1 & 1 & 0 & 0 \end{pmatrix}$$



### Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 struct nodo {
7     int info;
8     struct nodo *next;
9 };
10
11 int matriceCasuale(int A[MAX][MAX]) {
12     int n, i, j;
13     printf("Ordine della matrice: ");
14     scanf("%d", &n);
15     srand((unsigned)time(NULL));
16     for (i=0; i<n; i++)
    
```

```

17     for (j=0; j<n; j++)
18         A[i][j] = rand() % 3 - 1;
19     return n;
20 }
21
22 void stampaMatrice(int A[MAX][MAX], int n) {
23     int i, j;
24     for (i=0; i<n; i++) {
25         for (j=0; j<n; j++)
26             printf("%2d ", A[i][j]);
27         printf("\n");
28     }
29     return;
30 }
31
32 int adiacenti(struct nodo *V[], int i, int j) {
33     int rc;
34     struct nodo *p;
35     p = V[i];
36     while (p != NULL && p->info != j)
37         p = p->next;
38     if (p != NULL)
39         rc = 1;
40     else
41         rc = 0;
42     return rc;
43 }
44
45 void aggiungiSpigolo(struct nodo *V[], int i, int j) {
46     struct nodo *p;
47     p = malloc(sizeof(struct nodo));
48     p->info = j;
49     p->next = V[i];
50     V[i] = p;
51     return;
52 }
53
54 void creaGrafo(int A[MAX][MAX], struct nodo *V[], int n) {
55     int i, j;
56     for (i=0; i<n; i++)
57         V[i] = NULL;
58     for (i=0; i<n; i++) {
59         for (j=0; j<n; j++) {
60             if (A[i][j] > 0 && !adiacenti(V, i, j))
61                 aggiungiSpigolo(V, i, j);
62             else if (A[i][j] < 0 && !adiacenti(V, j, i))
63                 aggiungiSpigolo(V, j, i);
64         }
65     }
66     return;
67 }
68
69 void stampaLista(struct nodo *p) {
70     while (p != NULL) {
71         printf("%d --> ", p->info);
72         p = p->next;
73     }
74     printf("Null\n");

```

```

75     return;
76 }
77
78 void stampaGrafo(struct nodo *V[], int n) {
79     int i;
80     for (i=0; i<n; i++) {
81         printf("%2d: ", i);
82         stampaLista(V[i]);
83     }
84     return;
85 }
86
87 int main(void) {
88     int A[MAX][MAX], n;
89     struct nodo *V[MAX];
90     n = matriceCasuale(A);
91     printf("Matrice:\n");
92     stampaMatrice(A, n);
93     creaGrafo(A, V, n);
94     printf("\nListe di adiacenza del grafo:\n");
95     stampaGrafo(V, n);
96     return 0;
97 }

```

## Esercizio n. 2

Letti in input tre numeri interi  $n, h, k > 0$ , costruire due matrici di numeri interi casuali positivi minori di 100  $A$  e  $B$ , rispettivamente con  $n$  righe ed  $h$  colonne e con  $n$  righe e  $k$  colonne. Visualizzare in output entrambe le matrici. Visualizzare quindi la riga di indice  $i$  di entrambe le matrici, tali che

$$\left| \max_{0 \leq j \leq h} A_{i,j} - \min_{0 \leq j \leq k} B_{i,j} \right|$$

sia massimo.

**Esempio** Siano  $n = 5, h = 4, k = 6$  e si considerino le seguenti matrici:

$$A = \begin{pmatrix} 77 & 22 & 69 & 98 \\ 47 & 27 & 90 & 19 \\ 97 & 32 & 0 & 20 \\ 90 & 17 & 24 & 76 \\ 24 & 37 & 7 & 84 \end{pmatrix} \quad B = \begin{pmatrix} 53 & 89 & 56 & 81 & 15 & 68 \\ 27 & 4 & 63 & 15 & 32 & 86 \\ 33 & 36 & 39 & 94 & 7 & 28 \\ 31 & 9 & 41 & 63 & 83 & 93 \\ 94 & 34 & 34 & 1 & 3 & 2 \end{pmatrix}$$

Le righe che massimizzano la differenza tra l'elemento massimo di  $A$  e il minimo di  $B$  sono le due righe di indice 2 (la terza riga).

## Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #include <math.h>
5 #define MAX 100
6
7 void matriceCasuale(int X[MAX][MAX], int n, int m) {
8     int i, j;
9     for (i=0; i<n; i++)
10         for (j=0; j<m; j++)
11             X[i][j] = rand() % MAX;
12     return;
13 }
14
15 void stampaMatrice(int X[MAX][MAX], int n, int m) {
16     int i, j;
17     for (i=0; i<n; i++) {
18         for (j=0; j<m; j++)
19             printf("%2d ", X[i][j]);
20         printf("\n");
21     }
22     return;
23 }
24
25 int max(int X[MAX][MAX], int i, int h) {
26     int j, m = X[i][0];
27     for (j=1; j<h; j++)
28         if (X[i][j] > m)
29             m = X[i][j];
30     return m;
31 }
32
33 int min(int X[MAX][MAX], int i, int h) {
34     int j, m = X[i][0];
35     for (j=1; j<h; j++)
```

```

36     if (X[i][j] < m)
37         m = X[i][j];
38     return m;
39 }
40
41 int rigaMax(int A[MAX][MAX], int B[MAX][MAX], int n, int h, int k) {
42     int massimo=-1, iMassimo = -1, i;
43     for (i=0; i<n; i++) {
44         if (abs(max(A, i, h) - min(B, i, k)) > massimo) {
45             iMassimo = i;
46             massimo = abs(max(A, i, h) - min(B, i, k));
47         }
48     }
49     return iMassimo;
50 }
51
52 void stampaRiga(int X[MAX][MAX], int i, int m) {
53     int j;
54     for (j=0; j<m; j++)
55         printf("%2d ", X[i][j]);
56     printf("\n");
57     return;
58 }
59
60 int main(void) {
61     int A[MAX][MAX], B[MAX][MAX], n, h, k, i;
62     srand((unsigned)time(NULL));
63     printf("Inserisci tre interi positivi: ");
64     scanf("%d %d %d", &n, &h, &k);
65     matriceCasuale(A, n, h);
66     stampaMatrice(A, n, h);
67     matriceCasuale(B, n, k);
68     stampaMatrice(B, n, k);
69     i = rigaMax(A, B, n, h, k);
70     stampaRiga(A, i, h);
71     stampaRiga(B, i, k);
72     return 0;
73 }

```