

## Esame scritto del 28 Gennaio 2022 (Appello A)

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** (o un altro codice identificativo di fantasia) dello studente.

### Esercizio n. 1

Letti in input due numeri interi positivi  $n$ ,  $m$ , generare due matrici  $A$  e  $B$  di numeri interi positivi casuali minori di 100, di  $n$  righe e  $m$  colonne. Visualizzare in output le due matrici. Visualizzare il valore  $\min_{0 \leq i < n} (\max_{0 \leq j < m} |A_{i,j} - B_{i,j}|)$  e i valori degli indici  $i$  e  $j$  per cui si ottiene tale valore. Ricordiamo che in linguaggio C la funzione di libreria “abs(...)” restituisce il valore assoluto dell’argomento.

**Esempio** Siano  $n = 4$  e  $m = 5$  e si considerino le seguenti matrici di numeri interi casuali:

$$A = \begin{pmatrix} 45 & 7 & 27 & 20 & 34 \\ 2 & 99 & 76 & 5 & 87 \\ 44 & 63 & 30 & 44 & 49 \\ 7 & 15 & 79 & 56 & 74 \end{pmatrix} \quad B = \begin{pmatrix} 18 & 18 & 46 & 84 & 89 \\ 19 & 4 & 71 & 98 & 93 \\ 90 & 58 & 75 & 93 & 86 \\ 9 & 39 & 64 & 83 & 16 \end{pmatrix}$$

Il valore min-max cercato è 49 e si trova con gli elementi della terza riga e della quarta colonna.

### Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 50
5
6 void generaMatrice(int A[MAX][MAX], int n, int m) {
7     for (int i=0; i<n; i++)
8         for (int j=0; j<m; j++)
9             A[i][j] = rand() % 100;
10    return;
11 }
12
13 void stampaMatrice(int A[MAX][MAX], int n, int m) {
14     for (int i=0; i<n; i++) {
15         for (int j=0; j<m; j++)
16             printf("%2d ", A[i][j]);
17         printf("\n");
18     }
19     printf("\n");
20    return;
21 }
```

```

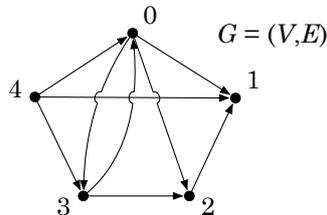
22
23 int main(void) {
24     int A[MAX][MAX], B[MAX][MAX], n, m, i, j, min=100, max, imin, jmax, jmaxtmp;
25     srand((unsigned)time(NULL));
26     printf("Numero di righe e colonne: ");
27     scanf("%d %d", &n, &m);
28     generaMatrice(A, n, m);
29     stampaMatrice(A, n, m);
30     generaMatrice(B, n, m);
31     stampaMatrice(B, n, m);
32     for (i = 0; i<n; i++) {
33         max = -1;
34         for (j=0; j<m; j++) {
35             if (abs(A[i][j] - B[i][j]) > max) {
36                 max = abs(A[i][j] - B[i][j]);
37                 jmaxtmp = j;
38             }
39         }
40         if (max < min) {
41             min = max;
42             imin = i;
43             jmax = jmaxtmp;
44         }
45     }
46     printf("Il valore min-max e' %d negli elementi (%d,%d)\n", min, imin, jmax);
47     return(0);
48 }

```

## Esercizio n. 2

Letto in input un grafo orientato  $G = (V, E)$  con  $n$  vertici, rappresentarlo con liste di adiacenza. Indichiamo con  $e_v$  il grado entrante di ciascun vertice  $v \in V(G)$ . Visualizzare in output una coppia di vertici  $u, v \in V(G)$  tale che  $u \neq v$  e  $|e_u - e_v|$  sia massimo.

**Esempio** Si consideri il grafo  $G = (V, E)$  rappresentato in figura.



La coppia di vertici con la massima differenza di grado entrante è data da  $u = 1$  con  $e_1 = 3$  e  $v = 4$  con  $e_4 = 0$ .

## Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 50
4
5 struct nodo {
6     int info;
7     struct nodo *next;
8 };
9
10 struct nodo *leggiLista(void) {
11     struct nodo *p, *primo=NULL;
12     int i, n;
13     printf("Numero di elementi: ");
14     scanf("%d", &n);
15     printf("Inserisci %d elementi: ", n);
16     for (i=0; i<n; i++) {
17         p = malloc(sizeof(struct nodo));
18         scanf("%d", &p->info);
19         p->next = primo;
20         primo = p;
21     }
22     return(primo);
23 }
24
25 int leggiGrafo(struct nodo *G[]) {
26     int i, n;
27     printf("Numero di vertici del grafo: ");
28     scanf("%d", &n);
29     for (i=0; i<n; i++) {
30         printf("Lista dei vertici adiacenti al vertice %d:\n", i);
31         G[i] = leggiLista();
32     }
33     return(n);
```

```

34 }
35
36 int grado(struct nodo *G[], int n, int x) {
37     struct nodo *p;
38     int i, g = 0;
39     for (i=0; i<n; i++) {
40         if (i != x) {
41             p = G[i];
42             while (p != NULL) {
43                 if (p->info == x)
44                     g++;
45                 p = p->next;
46             }
47         }
48     }
49     return(g);
50 }
51
52 int main(void) {
53     struct nodo *G[MAX];
54     int u, v, gu, gv, n, max = -1, max_u, max_v;
55     n = leggiGrafo(G);
56     for (u=0; u<n; u++) {
57         gu = grado(G, n, u);
58         for (v=0; v<n; v++) {
59             gv = grado(G, n, v);
60             if (u != v && abs(gu - gv) > max) {
61                 max_u = u;
62                 max_v = v;
63                 max = abs(gu - gv);
64             }
65         }
66     }
67     printf("La coppia di vertici e' %d e %d\n", max_u, max_v);
68     return(0);
69 }

```