

Algoritmi e Strutture Dati (IN110)

Esercitazione n. 6

Marco Liverani *

Esercizio n. 1

Generare una matrice M di $n \times n$ numeri casuali. Leggere in input una sequenza X di $2k$ numeri interi tali che $X_i \in \{-1, 0, 1\} \forall i = 0, 1, \dots, 2k - 1$. Partendo dall'elemento $M_{0,0}$ della matrice, ogni coppia di elementi di X , $(x_i, x_{i+1}), i = 0, 2, 4, \dots, 2k - 2$, rappresenta uno spostamento sulla matrice M .

Stampare l'elemento su cui si ferma la sequenza di spostamenti X dopo le k "mosse". Se una delle mosse è "proibita" perché porterebbe ad uscire dalla matrice M , allora deve essere stampato un messaggio di errore ed il programma deve interrompere la sua esecuzione.

Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX_N 20
5 #define MAX_X 30
6
7 void stampa_matrice(int M[MAX_N][MAX_N], int n, int m) {
8     int i, j;
9     for (i=0; i<n; i++) {
10         for (j=0; j<m; j++)
11             printf("%2d ", M[i][j]);
12         printf("\n");
13     }
14     return;
15 }
16
17 int genera_matrice(int M[MAX_N][MAX_N]) {
18     int n, i, j;
19     srand((unsigned)time(NULL));
20     n = rand() % MAX_N;
21     for (i=0; i<n; i++)
22         for (j=0; j<n; j++)
23             M[i][j] = rand() % 100;
24     return(n);
25 }
26
```

*Università degli Studi Roma Tre, Corso di Laurea in Matematica, Corso di Algoritmi e Strutture Dati (IN110) – sito web del corso <http://www.mat.uniroma3.it/users/liverani/IN110/>

```

27 int leggi_array(int X[]) {
28     int n, i;
29     printf("numero di mosse: ");
30     scanf("%d", &n);
31     printf("inserisci le %d coppie di numeri che rappresentano le mosse: ", n);
32     n = n*2;
33     for (i=0; i<n; i++)
34         scanf("%d", &X[i]);
35     return(n);
36 }
37
38 int main(void) {
39     int M[MAX_N][MAX_N], X[MAX_X];
40     int k, n, m, i, x=0, y=0;
41
42     n = genera_matrice(M);
43     printf("E' stata generata una matrice %dx%d.\n", n, n);
44     stampa_matrice(M, n, n);
45     m = leggi_array(&X[0]);
46
47     for (i=0; i<m && x+X[i]>=0 && x+X[i]<n && y+X[i+1]>=0 && y+X[i+1]<n; i+=2) {
48         x += X[i];
49         y += X[i+1];
50     }
51     if (i==m) {
52         printf("Il cammino sulla matrice e' terminato sull'elemento ");
53         printf("M[%d][%d] = %d.\n", x, y, M[x][y]);
54     } else {
55         printf("Errore: il cammino esce dai limiti della matrice!\n");
56     }
57     return(1);
58 }

```

Esercizio n. 2

Leggere in input due sequenze di n ed m numeri *floating point* e memorizzarle in due array A e B . Senza utilizzare un terzo array di appoggio, inserire gli elementi di B al centro dell'array A e stampare A .

Ad esempio, siano $A = (3, 2, 4, 7, 6, 5)$ e $B = (13, 1, 9)$. Allora al termine dell'elaborazione si avrà $A = (3, 2, 4, 13, 1, 9, 7, 6, 5)$.

Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 int leggi_array(float x[]) {
6     int n, i;
7     scanf("%d", &n);
8     for (i=0; i<n; i++)
9         scanf("%f", &x[i]);
10    return(n);
11 }
12
13 void stampa_array(float x[], int n) {
14     int i;
15     for (i=0; i<n; i++)
16         printf("%f ", x[i]);
17     printf("\n");
18     return;
19 }
20
21 void inserisci(float x[], float y[], int n, int m) {
22     int i, j=0;
23     for (i=n-1; i>=n/2; i--)
24         x[i+m] = x[i];
25     for (i=n/2; i<n/2+m; i++)
26         x[i] = y[i - n/2];
27     return;
28 }
29
30 int main(void) {
31     float A[MAX], B[MAX];
32     int n, m;
33     n = leggi_array(A);
34     m = leggi_array(B);
35     inserisci(A, B, n, m);
36     stampa_array(A, n+m);
37     return(1);
38 }
```