

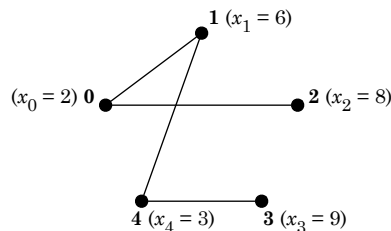
## Esame scritto del 22 gennaio 2010

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. Si richiede di riportare sul foglio del compito il proprio nominativo completo ed il numero di matricola o un codice identificativo personale equivalente.

### Esercizio n. 1

Letti in input due interi  $n, k > 0$  generare una sequenza di  $n$  interi casuali  $X = (x_0, x_1, \dots, x_{n-1})$  tale che  $0 < x_i \leq k$  per ogni  $i = 0, 1, \dots, n - 1$ . Stampare la sequenza  $X$ . Costruire le liste di adiacenza del grafo non orientato  $G = (V, E)$  tale che  $V = \{0, 1, \dots, n - 1\}$  ed  $E = \{(i, j) : x_i = hx_j \text{ per qualche } h \in \mathbb{N}\}$ . Stampare le liste di adiacenza di  $G$ .

**Esempio** Sia  $n = 5, k = 9$ ; si consideri la sequenza di interi positivi:  $V = (2, 6, 8, 9, 3)$ . Il grafo  $G$  di cui devono essere prodotte le liste di adiacenza è rappresentato nella figura seguente.



### 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 sequenzaCasuale(int A[]) {
12     int i, n, k;
13     printf("Numero di elementi: ");
14     scanf("%d", &n);

```

```

15 | printf("Valore soglia: ");
16 | scanf("%d", &k);
17 | srand((unsigned)time(NULL));
18 | for (i=0; i<n; i++)
19 |     A[i] = rand() % k +1;
20 | return(n);
21 | }
22 |
23 | void costruisciGrafo(struct nodo *G[], int n, int X[]) {
24 |     int i, j;
25 |     struct nodo *p;
26 |     for (i=0; i<n; i++)
27 |         G[i] = NULL;
28 |     for (i=0; i<n-1; i++)
29 |         for (j=i+1; j<n; j++)
30 |             if (X[i] % X[j] == 0 || X[j] % X[i] == 0) {
31 |                 p = malloc(sizeof(struct nodo));
32 |                 p->info = i;
33 |                 p->next = G[j];
34 |                 G[j] = p;
35 |                 p = malloc(sizeof(struct nodo));
36 |                 p->info = j;
37 |                 p->next = G[i];
38 |                 G[i] = p;
39 |             }
40 |     return;
41 | }
42 |
43 | void stampaLista(struct nodo *p) {
44 |     while (p != NULL) {
45 |         printf("%d --> ", p->info);
46 |         p = p->next;
47 |     }
48 |     printf("NULL\n");
49 |     return;
50 | }
51 |
52 | void stampaGrafo(struct nodo *G[], int n) {
53 |     int i;
54 |     for (i=0; i<n; i++) {
55 |         printf("%d: ", i);
56 |         stampaLista(G[i]);
57 |     }
58 |     return;
59 | }
60 |
61 | void stampaVettore(int A[], int n) {
62 |     int i;
63 |     for (i=0; i<n; i++)
64 |         printf("%d ", A[i]);
65 |     printf("\n");
66 |     return;
67 | }
68 |

```

```

69 int main(void) {
70     int n, X[MAX];
71     struct nodo *G[MAX];
72     n = sequenzaCasuale(X);
73     printf("Sequenza di numeri casuali: ");
74     stampaVettore(X, n);
75     costruisciGrafo(G, n, X);
76     stampaGrafo(G, n);
77     return(0);
78 }

```

## Esercizio n. 2

Leggere in input due matrici di numeri interi  $A$  e  $B$ , rispettivamente di ordine  $n \times m$  e  $m \times k$ . Costruire il vettore  $C$  di cardinalità  $m$  definito nel modo seguente:  $C_i = \max\{\min A^{(i)}, \min B_{(i)}\}$ ,  $i = 0, 1, \dots, m - 1$ . Con  $A^{(i)}$  e  $B_{(i)}$  si sono indicate rispettivamente la colonna di indice  $i$  di  $A$  e la riga di indice  $i$  di  $B$ .

**Esempio** Si considerino le seguenti matrici  $A$  (di ordine  $4 \times 5$ ) e  $B$  (di ordine  $5 \times 3$ ):

$$A = \begin{pmatrix} 29 & \mathbf{13} & 7 & 94 & 20 \\ 11 & 253 & 19 & 72 & 17 \\ \mathbf{8} & 24 & 59 & \mathbf{46} & 85 \\ 74 & 39 & 69 & 74 & \mathbf{2} \end{pmatrix} \quad B = \begin{pmatrix} 71 & \mathbf{42} & 49 \\ 24 & 69 & \mathbf{23} \\ \mathbf{5} & 11 & 124 \\ 82 & \mathbf{17} & 31 \\ \mathbf{32} & 47 & 88 \end{pmatrix}$$

Il vettore  $C$  è il seguente:  $C = (42, 23, 7, 46, 32)$ .

## Soluzione

```

1  #include <stdlib.h>
2  #include <stdio.h>
3  #define MAX 100
4
5  void calcolaVettore(int A[][MAX], int B[][MAX], int C[], int n, int m, int k) {
6      int x, y, i, h;
7      for (h = 0; h < m; h++) {
8          x = A[0][h];
9          for (i = 1; i < n; i++)
10             if (A[i][h] < x)
11                 x = A[i][h];
12         y = B[h][0];
13         for (i = 1; i < k; i++)
14             if (B[h][i] < y)
15                 y = B[h][i];
16         if (x > y)
17             C[h] = x;
18         else
19             C[h] = y;
20     }
21     return;
22 }

```

```

23
24 void stampaVettore(int A[], int n) {
25     int i;
26     for (i=0; i<n; i++)
27         printf("%d ", A[i]);
28     printf("\n");
29     return;
30 }
31
32 void leggiMatrice(int X[][MAX], int *n, int *m) {
33     int i, j;
34     printf("Numero di righe: ");
35     scanf("%d", n);
36     printf("Numero di colonne: ");
37     scanf("%d", m);
38     for (i=0; i<*n; i++) {
39         printf("Elementi della riga %d: ", i);
40         for (j=0; j<*m; j++)
41             scanf("%d", &X[i][j]);
42     }
43     return;
44 }
45
46 int main(void) {
47     int n, m, k, A[MAX][MAX], B[MAX][MAX], C[MAX];
48     leggiMatrice(A, &n, &m);
49     leggiMatrice(B, &m, &k);
50     calcolaVettore(A, B, C, n, m, k);
51     stampaVettore(C, m);
52     return(0);
53 }

```