

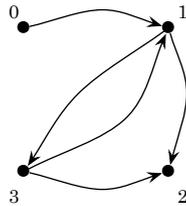
Esame scritto del 20 gennaio 2011

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

Leggere in input un grafo orientato $G = (V, E)$ e memorizzarlo per mezzo di liste di adiacenza. Letta in input una lista con un sottoinsieme V' di vertici di V , calcolare la somma dei gradi entranti dei vertici di V' .

Esempio Sia $n = 4$ e sia G il grafo riportato in figura; se $V' = \{1, 3\}$ allora la somma dei gradi entranti dei vertici di V' è 3.



Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 100
4
5 struct nodo {
6     int info;
7     struct nodo *next;
8 };
9
10 int grado(struct nodo *G[], int n, int v) {
11     struct nodo *p;
12     int x=0, i;
13     for (i=0; i<n; i++) {
14         if (i != v) {
15             p = G[i];
16             while (p != NULL && p->info != v)
17                 p = p->next;
18             if (p != NULL)
19                 x++;
        }
    }
}
  
```

```

20     }
21 }
22 return(x);
23 }
24
25 struct nodo *leggiLista(void) {
26     struct nodo *p, *primo = NULL;
27     int i, n;
28     printf("Numero di elementi: ");
29     scanf("%d", &n);
30     printf("Inserisci %d vertici: ", n);
31     for (i=0; i<n; i++) {
32         p = malloc(sizeof(struct nodo));
33         scanf("%d", &p->info);
34         p->next = primo;
35         primo = p;
36     }
37     return(primo);
38 }
39
40 int leggiGrafo(struct nodo *G[]) {
41     int i, n;
42     printf("Numero di vertici: ");
43     scanf("%d", &n);
44     for (i=0; i<n; i++) {
45         printf("Lista di adiacenza di %d\n", i);
46         G[i] = leggiLista();
47     }
48     return(n);
49 }
50
51 int main(void) {
52     struct nodo *p, *G[MAX];
53     int n, s = 0;
54     n = leggiGrafo(G);
55     p = leggiLista();
56     while (p != NULL) {
57         s = s + grado(G, n, p->info);
58         p = p->next;
59     }
60     printf("Somma dei gradi entranti di V': %d.\n", s);
61     return(0);
62 }

```

Esercizio n. 2

Letti in input due numeri interi positivi n e m , generare una matrice A di numeri interi casuali compresi tra -7 e 31 , estremi inclusi, con n righe ed m colonne. Verificare se esiste una riga della matrice la cui somma degli elementi sia uguale alla somma degli elementi di una colonna; in caso positivo stampare la riga e la colonna.

Esempio Sia $n = 3$ e $m = 4$ e si consideri la seguente matrice:

$$A = \begin{pmatrix} 1 & -3 & 4 & 29 \\ -4 & -1 & 18 & 2 \\ 14 & 12 & -7 & 6 \end{pmatrix}$$

La terza colonna e la seconda riga hanno la stessa somma (15).

Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 void generaMatrice(int A[MAX][MAX], int n, int m) {
7     int i, j;
8     srand((unsigned)time(NULL));
9     for (i=0; i<n; i++)
10         for (j=0; j<m; j++)
11             A[i][j] = rand() % 39 - 7;
12     return;
13 }
14
15 void verifica(int A[MAX][MAX], int n, int m) {
16     int i, j, k, sr, sc, flag = 0;
17     for (i=0; i<n && flag == 0; i++) {
18         sr = 0;
19         for (j=0; j<m; j++)
20             sr = sr + A[i][j];
21         for (j=0; j<m && flag == 0; j++) {
22             sc = 0;
23             for (k=0; k<n; k++)
24                 sc = sc + A[k][j];
25             if (sc == sr) {
26                 flag = 1;
27                 printf("La riga %d ha somma uguale alla colonna %d (%d).\n", i, j, sc);
28             }
29         }
30     }
31     if (flag == 0)
32         printf("Non esistono righe e colonne con la stessa somma.\n");
33     return;
34 }
35
36 void stampaMatrice(int A[MAX][MAX], int n, int m) {
37     int i, j;
```

```
38  for (i=0; i<n; i++) {
39      for (j=0; j<m; j++)
40          printf("%2d ", A[i][j]);
41          printf("\n");
42      }
43      return;
44  }
45
46  int main(void) {
47      int n, m, A[MAX][MAX];
48      printf("Numero di righe e di colonne: ");
49      scanf("%d %d", &n, &m);
50      generaMatrice(A, n, m);
51      stampaMatrice(A, n, m);
52      verifica(A, n, m);
53      return(0);
54  }
```