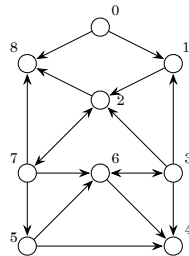


## Esame scritto del 26 gennaio 2009

### Esercizio n. 1

Letto in input un grafo orientato, rappresentarlo mediante liste di adiacenza. Stampare tutti i vertici di grado entrante massimo.

**Esempio** Si consideri il grafo in figura. I vertici di grado entrante massimo sono 2, 4, 6 e 8, tutti con grado entrante pari a 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 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         p->next = primo;
19         scanf("%d", &p->info);
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 di adiacenza del vertice %d.\n", i);
31         G[i] = leggiLista();
32     }
33     return(n);
34 }
35
36 void gradiEntranti(struct nodo *G[ ], int n, int d[ ]) {
37     struct nodo *p;
38     int i;
39     for (i=0; i<n; i++)
40         d[i] = 0;
41     for (i=0; i<n; i++) {
42         p = G[i];
43         while (p != NULL) {
44             d[p->info]++;
45             p = p->next;
46         }
47     }
48     return;
49 }
50
51 int max(int v[ ], int n) {
52     int m, i;
53     m = v[0];
54     for (i=1; i<n; i++)
55         if (v[i] > m)
56             m = v[i];
57     return(m);
58 }
59
60 int main(void) {
61     int n, m, i, d[MAX];
62     struct nodo *G[MAX];
63     n = leggiGrafo(G);
64     gradiEntranti(G, n, d);
65     m = max(d, n);
66     printf("Il grado entrante massimo per i vertici del grafo e' %d.\n", m);
67     printf("I vertici con grado entrante massimo sono i seguenti: ");
68     for (i=0; i<n; i++)
69         if (d[i] == m)
70             printf("%d ", i);
71     printf("\n");
72     return(0);
73 }

```

## Esercizio n. 2

Leggere in input un numero intero positivo  $n$  e una matrice  $C$  di  $n \times n$  caratteri. Leggere in input un intero positivo  $k$  e una sequenza di  $2k$  numeri interi e memorizzarla in un vettore  $V$ . La sequenza  $V_0, V_1, \dots, V_{2k-1}$  rappresenta un messaggio criptato da decifrare utilizzando la matrice  $C$ . Ogni coppia  $V_{2i}, V_{2i+1}$ , per  $i = 0, 1, 2, \dots, k-1$  indica il numero di riga e il numero di colonna dell'elemento della matrice  $C$  che contiene il carattere corrispondente del messaggio. Costruire una stringa con il messaggio "in chiaro" e stamparla in output.

**Esempio** Sia  $n = 5$ ,  $k = 4$ . Supponiamo di aver letto in input la seguente matrice:

$$C = \begin{pmatrix} a & b & c & d & e \\ h & q & r & p & a \\ k & s & m & g & z \\ y & u & o & n & d \\ t & v & x & e & l \end{pmatrix}$$

Il messaggio da decifrare è il seguente:  $V = (0, 2, 1, 4, 3, 3, 0, 4)$ . Il messaggio in chiaro è la parola "cane".

## Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 50
4
5 int leggiMatrice(char A[MAX][MAX]) {
6     int i, j, n;
7     char s[MAX];
8     printf("Ordine della matrice quadrata: ");
9     scanf("%d", &n);
10    for (i=0; i<n; i++) {
11        printf("Inserisci i caratteri della riga %d: ", i);
12        scanf("%s", s);
13        for (j=0; j<n; j++)
14            A[i][j] = s[j];
15        fflush(stdin);
16    }
17    return(n);
18 }
19
20 void stampaMatrice(char A[MAX][MAX], int n) {
21     int i, j;
22     for (i=0; i<n; i++) {
23         for (j=0; j<n; j++)
24             printf("%c ", A[i][j]);
25         printf("\n");
26     }
27     printf("\n");
28     return;
29 }
30
31 int leggiVettore(int A[ ]) {
32     int i, n;
33     printf("Numero di caratteri del messaggio da decifrare: ");
34     scanf("%d", &n);
35     printf("Inserisci le %d coppie di interi del messaggio cifrato: ", n);
36     for (i=0; i<2*n; i++)
37         scanf("%d", &A[i]);
38     return(n);
39 }
40
41 void decifra(char C[MAX][MAX], int V[MAX], int k, char s[ ]) {
42     int i;
43     for (i=0; i<k; i++)
44         s[i] = C[V[2*i]][V[2*i+1]];
45     s[i] = '\0';
```

```
46     return;  
47 }  
48  
49 int main(void) {  
50     char C[MAX][MAX], s[MAX];  
51     int V[MAX], k, n;  
52     k = leggiVettore(V);  
53     n = leggiMatrice(C);  
54     decifra(C, V, k, s);  
55     printf("Il messaggio in chiaro e' il seguente: '%s'\n", s);  
56     return(0);  
57 }
```