

## Esame scritto del 7 settembre 2007

### Esercizio n. 1

Letta una stringa  $S$  di caratteri, memorizzarla in un vettore. Riportare sulle righe di una matrice  $M$  le sottostringhe di  $S$  separate da uno o più asterischi. Stampare la matrice  $M$ .

**Esempio** Si consideri la stringa  $S = \text{"pane**pasta*biscotti**uova*prosciutto"}$ . La matrice  $M$  da produrre è la seguente:

|   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|
| p | a | n | e |   |   |   |   |   |   |  |  |  |  |  |
| p | a | s | t | a |   |   |   |   |   |  |  |  |  |  |
| b | i | s | c | o | t | t | i |   |   |  |  |  |  |  |
| u | o | v | a |   |   |   |   |   |   |  |  |  |  |  |
| p | r | o | s | c | i | u | t | t | o |  |  |  |  |  |
|   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |

### Soluzione

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#define MAX 100

void stampa_matrice(char M[MAX][MAX], int n) {
    int i;

    for (i=0; i<n; i++)
        printf("%s\n", &M[i][0]);
    return;
}

int main(void) {
    char s[MAX], M[MAX][MAX];
    int k=0, j=0, n=0;

    printf("Inserisci una stringa di caratteri: ");
    scanf("%s", s);
    while (s[k] != '\0') {
        j = 0;
        while (s[k] != '*' && s[k] != '\0') {
            M[n][j] = s[k];
            j++;
            k++;
        }
        M[n][j] = '\0';
        while (s[k] == '*')
            k++;
        if (j > 0)
            n++;
    }
}
```

```

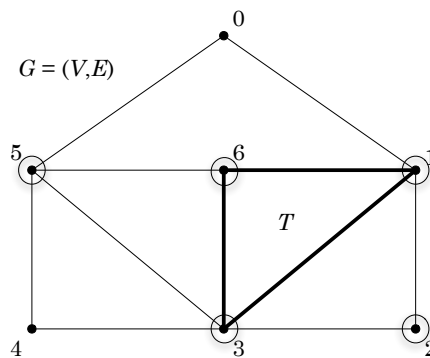
    stampa_matrice(M, n);
    return(0);
}

```

## Esercizio n. 2

In un grafo  $G$  non orientato un triangolo è un ciclo di lunghezza 3; dato un triangolo  $T$  di  $G$ , un *triangolo esteso* di  $T$  è il sottografo di  $G$  indotto dai vertici di  $G$  adiacenti ad almeno due vertici di  $T$ . Letto un grafo non orientato  $G$  rappresentarlo con liste di adiacenza. Letti in input tre vertici del grafo,  $u$ ,  $v$  e  $w$ , verificare che tali vertici costituiscano un triangolo  $T$  in  $G$ . In caso positivo, stampare i vertici che formano il triangolo esteso di  $T$ .

**Esempio** Si consideri il grafo  $G$  rappresentato in figura e siano  $u = 1, v = 3, w = 6$  i tre vertici letti in input che costituiscono il triangolo  $T$ . Il triangolo esteso di  $T$  è il sottografo di  $G$  indotto dai vertici 1, 2, 3, 5, 6, evidenziati in figura con un cerchietto.



## Soluzione

```

#include <stdlib.h>
#include <stdio.h>
#define MAX 50

struct nodo {
    int info;
    struct nodo *next;
};

struct nodo *leggi_lista(void) {
    struct nodo *p, *primo=NULL;
    int i, n;
    printf("Numero di elementi: ");
    scanf("%d", &n);
    printf("Inserisci %d elementi: ", n);
    for (i=0; i<n; i++) {
        p = malloc(sizeof(struct nodo));
        scanf("%d", &p->info);
        p->next = primo;
        primo = p;
    }
    return(primo);
}

```

```

int leggi_grafo(struct nodo *G[]) {
    int i, n;
    printf("Numero di vertici del grafo: ");
    scanf("%d", &n);
    for (i=0; i<n; i++) {
        printf("Lista dei vertici adiacenti al vertice %d:\n", i);
        G[i] = leggi_lista();
    }
    return(n);
}

int adiacenti(struct nodo *G[], int u, int v) {
    int risp;
    struct nodo *p;
    p = G[u];
    while (p != NULL && p->info != v)
        p = p->next;
    if (p == NULL)
        risp = 0;
    else
        risp = 1;
    return(risp);
}

void triangolo_esteso(struct nodo *G[], int n, int u, int v, int w) {
    struct nodo *p;
    int i, cont;
    printf("Il triangolo esteso di T e' dato dai seguenti vertici: {");
    for (i=0; i<n; i++) {
        p = G[i];
        cont = 0;
        while (p != NULL) {
            if (p->info == u || p->info == v || p->info == w)
                cont++;
            p = p->next;
        }
        if (cont >= 2)
            printf("%d ", i);
    }
    printf("}\n");
    return;
}

int main(void) {
    struct nodo *G[MAX];
    int n, u, v, w;
    n = leggi_grafo(G);
    printf("Inserisci tre vertici di G: ");
    scanf("%d %d %d", &u, &v, &w);
    if (adiacenti(G, u, v) && adiacenti(G, v, w) && adiacenti(G, u, w)) {
        printf("\nIl sottografo indotto da T={%d,%d,%d} e' un triangolo.\n", u, v, w);
        triangolo_esteso(G, n, u, v, w);
    } else {
        printf("\nIl sottografo indotto da T={%d,%d,%d} non e' un triangolo.\n", u, v, w);
    }
    return(0);
}

```