

Esame scritto del 21 giugno 2007

Esercizio n. 1

Lette due stringhe di caratteri alfanumerici s e t , verificare quante volte la stringa t è contenuta in s .

Esempio Si considerino le seguenti stringhe:

$$s = \text{"ABACAABBAABAC"}, t = \text{"ABA"}$$

La stringa t è contenuta in s due volte:

$$s = \text{"\overbrace{\text{ABA}}^t \text{ CAABBA } \overbrace{\text{ABA}}^t \text{ C"}$$

Soluzione

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

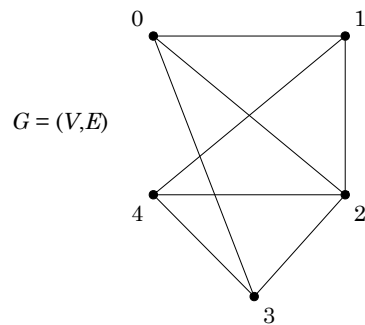
int main(void) {
    int i, j, cont=0;
    char s[MAX], t[MAX];

    printf("Inserisci le due stringhe s e t: ");
    scanf("%s %s", s, t);
    for (i=0; i<=strlen(s)-strlen(t); i++) {
        j = 0;
        while (j<strlen(t) && s[i+j] == t[j])
            j++;
        if (j==strlen(t))
            cont++;
    }
    printf("s='%s' contiene t='%s' %d volte.\n", s, t, cont);
    return(0);
}
```

Esercizio n. 2

Letto un grafo $G = (V, E)$ non orientato rappresentarlo con liste di adiacenza. Stampare, se esistono, gli insiemi di vertici gemelli. Ricordiamo che due vertici $u, v \in V(G)$ si dicono *gemelli* se $N(u) = N(v)$, dove $N(x) = \{y \in V(G) : (x, y) \in E(G)\}$.

Esempio Si consideri il grafo G rappresentato in figura. Gli insiemi di vertici gemelli sono due: $A_1 = \{0, 4\}$ e $A_2 = \{1, 3\}$. Il vertice 2 non è gemello di nessun altro vertice del grafo.



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 contiene(struct nodo *u, struct nodo *v) {
    int risp=1;
    struct nodo *p;
    while (u != NULL && risp == 1) {
        p = v;
        while (p != NULL && p->info != u->info)
  
```

```

        p = p->next;
        if (p == NULL)
            risp = 0;
        u = u->next;
    }
    return(risp);
}

int main(void) {
    struct nodo *G[MAX];
    int n, i, j, x[MAX];

    n = leggi_grafo(G);
    for (i=0; i<n; i++)
        x[i] = -1;
    for (i=0; i<n-1; i++) {
        if (x[i] == -1) {
            for (j=i+1; j<n; j++) {
                if (x[j] == -1 && contiene(G[i], G[j]) && contiene(G[j], G[i])) {
                    x[i] = i;
                    x[j] = i;
                }
            }
        }
    }
    printf("Insiemi di vertici gemelli:\n");
    for (i=0; i<n; i++) {
        if (x[i] == i) {
            printf("%d", i);
            for (j=i+1; j<n; j++) {
                if (x[j] == i)
                    printf(", %d", j);
            }
            printf("}\n");
        }
    }
    return(0);
}

```