

Per evitare di cadere in un ciclo infinito, ogni nodo del percorso deve essere visitato una e una sola volta. A questo scopo si usa un array (`gia_visitato[]`) che contiene tanti elementi interi quanti sono i nodi del grafo memorizzati nell'array `s[]` e nello stesso ordine di `s[]`. Tutto l'array viene inizializzato a zero, prima di transitare per un certo nodo ci si domanda se l'elemento corrispondente di `gia_visitato` ha valore uno, se così non è si assegna uno a quell'elemento e lo si visita.

Devono essere dichiarate le seguenti funzioni e variabili globali:

```

void giro( char );
void vai( int );

int gia_visitati[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

main()
{
char etichetta, invio;

mat_adiacenze();
vis_mat_adiacenze();
successori();
visita();

/* VISITA A PARTIRE DA UN NODO */
printf("\nImmettere l'etichetta di un nodo: ");
scanf("%c", &invio);
scanf("%c", &etichetta);
giro( etichetta );
}

void giro(char e)
{
int i = 0;
printf("\n");
while(i<n && s[i].inf!=e) i++;
if (i==n) return;
vai(i);
}

void vai(int i)
{
struct successore *p;

p = s[i].pun;
while(p!=NULL) {
    if(gia_visitati[p->inf]==0) {
        gia_visitati[p->inf] = 1;
        printf(" %c", s[p->inf].inf);
        vai( p->inf );
    }
    p = p->pun;
}
}

```