

Esame scritto del 10 settembre 2008

Esercizio n. 1

Leggere in input un numero intero positivo $x < 100.000$. Costruire una matrice A di caratteri con 5 righe e 10 colonne sulla base del seguente procedimento. Sia y la i -esima cifra di x (partendo da destra). Assegnare ai primi y elementi della i -esima riga di A il carattere “*”; assegnare a tutte le rimanenti celle di A il carattere “-”. Stampare A .

Esempio Sia $x = 7023$. Allora la matrice A sarà la seguente:

$$A = \begin{pmatrix} * & * & * & - & - & - & - & - & - & - \\ * & * & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - \\ * & * & * & * & * & * & * & - & - & - \\ - & - & - & - & - & - & - & - & - & - \end{pmatrix}$$

Soluzione

```

1  #include <stdlib.h>
2  #include <stdio.h>
3
4  void costruisciMatrice(char A[5][10], long int x) {
5      int i, j, y;
6      for (i=0; i<5; i++) {
7          y = x - (x/10)*10;
8          for (j=0; j<y; j++)
9              A[i][j] = '*';
10         for (j=y; j<10; j++)
11             A[i][j] = '-';
12         x = x/10;
13     }
14     return;
15 }
16
17 void stampaMatrice(char A[5][10], int n, int m) {
18     int i, j;
19     for (i=0; i<n; i++) {
20         for (j=0; j<m; j++)
21             printf("%c ", A[i][j]);
22         printf("\n");
23     }
24     printf("\n");
25     return;
26 }
27
28 int main(void) {
29     char A[5][10];
30     long int x;
31     printf("Inserisci un intero: ");

```

```

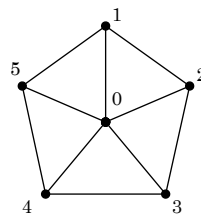
32 scanf("%ld", &x);
33 costruisciMatrice(A, x);
34 stampaMatrice(A, 5, 10);
35 return(0);
36 }

```

Esercizio n. 2

Letto in input un intero $n > 3$ costruire le liste di adiacenza di un grafo “wheel” non orientato con $n - 1$ raggi. Stampare le liste di adiacenza del grafo.

Esempio Sia $n = 6$. Il grafo wheel con 5 raggi è rappresentato nella seguente figura:



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 void stampaLista(struct nodo *p) {
11     while (p != NULL) {
12         printf("%d --> ", p->info);
13         p = p->next;
14     }
15     printf("NULL\n");
16     return;
17 }
18
19 void stampaGrafo(struct nodo *G[ ], int n) {
20     int i;
21     for (i=0; i<n; i++) {
22         printf("%3d: ", i);
23         stampaLista(G[i]);
24     }
25     return;
26 }
27
28 void aggiungiSpigoloOrientato(struct nodo *G[ ], int u, int v) {
29     struct nodo *p;
30     p = malloc(sizeof(struct nodo));
31     p->info = v;

```

```

32     p->next = G[u];
33     G[u] = p;
34     return;
35 }
36
37 void aggiungiSpigolo(struct nodo *G[ ], int u, int v) {
38     aggiungiSpigoloOrientato(G, u, v);
39     aggiungiSpigoloOrientato(G, v, u);
40     return;
41 }
42
43 void costruisciGrafo(struct nodo *G[ ], int n) {
44     int i;
45     for (i=0; i<n; i++)
46         G[i] = NULL;
47     for (i=1; i<n-1; i++) {
48         aggiungiSpigolo(G, 0, i);
49         aggiungiSpigolo(G, i, i+1);
50     }
51     aggiungiSpigolo(G, 0, n-1);
52     aggiungiSpigolo(G, n-1, 1);
53     return;
54 }
55
56 int main(void) {
57     int n;
58     struct nodo *G[MAX];
59     printf("inserisci un intero: ");
60     scanf("%d", &n);
61     costruisciGrafo(G, n);
62     stampaGrafo(G, n);
63     return(0);
64 }

```