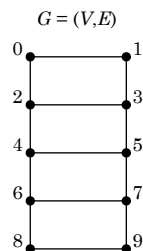


Esame scritto del 6 febbraio 2008

Esercizio n. 1

Letto in input un numero intero $k > 0$ costruire le liste di adiacenza di un grafo “scala” $G = (V, E)$ non orientato con $2k$ vertici. Stampare le liste di adiacenza del grafo G .

Esempio Un grafo “scala” con $2k$ vertici è costituito da una griglia rettangolare di lati 1 e k , come il grafo rappresentato in figura con $k = 5$. La numerazione dei vertici è del tutto arbitraria.



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 aggiungiSpigolo(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     p = malloc(sizeof(struct nodo));
35     p->info = u;
36     p->next = G[v];
37     G[v] = p;
38     return;
39 }
40
41 void costruisciScala(struct nodo *G[], int k) {
42     int i;
43     for (i=0; i<2*k; i++)
44         G[i] = NULL;
45     for (i=0; i<k; i++) {
46         aggiungiSpigolo(G, 2*i, 2*i+1);
47         if (i>0) {
48             aggiungiSpigolo(G, 2*i, 2*(i-1));
49             aggiungiSpigolo(G, 2*i+1, 2*i-1);
50         }
51     }
52     return;
53 }
54
55 int main(void) {
56     int k;
57     struct nodo *G[MAX];
58     printf("inserisci un intero: ");
59     scanf("%d", &k);
60     costruisciScala(G, k);
61     stampaGrafo(G, 2*k);
62     return(0);
63 }

```

Esercizio n. 2

Costruire un array A di n numeri interi casuali compresi tra 1 e 20. Stampare il vettore A . Per ogni elemento A_i dispari, eliminare dal vettore i successivi A_i elementi effettuando uno *shift* a sinistra degli elementi seguenti, compreso A_i . Stampare il vettore risultante.

Esempio Sia $n = 12$ e $A = (4, 16, 3, 2, 8, 14, 1, 6, 18, 7, 2, 20)$; in output viene prodotto il seguente array: $A = (4, 16, 14, 6, 18)$.

Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 int generaVettore(int V[], int min, int max) {
7     int i, n;
8     srand((unsigned)time(NULL));

```

```

9   printf("Numero di elementi: ");
10  scanf("%d", &n);
11  for (i=0; i<n; i++)
12    V[i] = (rand() % (max-min+1)) + min;
13  return(n);
14 }
15
16 void stampaVettore(int V[], int n) {
17     int i;
18     for (i=0; i<n; i++)
19         printf("%d ", V[i]);
20     printf("\n");
21     return;
22 }
23
24 int riduciVettore(int V[], int n) {
25     int i, j, k;
26     for (i=0; i<n; i++) {
27         if (V[i] % 2 == 1) {
28             if (i + V[i] > n) {
29                 n = i;
30             } else {
31                 k = V[i];
32                 for (j=i; j<n-k; j++)
33                     V[j] = V[j+k];
34                 n = n-k;
35                 stampaVettore(V, n);
36                 i--;
37             }
38         }
39     }
40     return(n);
41 }
42
43 int main(void) {
44     int A[MAX], n;
45     n = generaVettore(A, 1, 20);
46     stampaVettore(A, n);
47     n = riduciVettore(A, n);
48     stampaVettore(A, n);
49     return(0);
50 }

```