

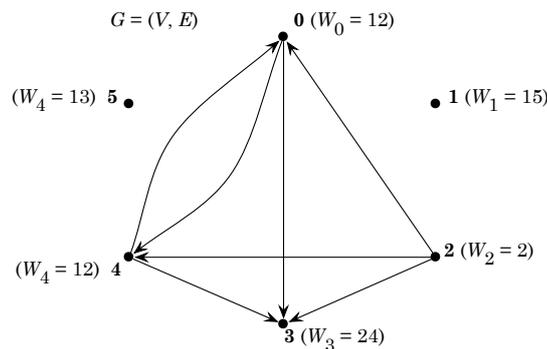
Esame scritto del 8 Febbraio 2013

Si richiede di risolvere entrambi gli esercizi riportando una codifica in linguaggio C completa dei due programmi. Nel caso in cui non si riesca a completare entrambi gli esercizi si suggerisce di riportare almeno la codifica in C delle funzioni principali o una loro pseudo-codifica. È possibile consultare libri e appunti personali, ma non scambiare libri o appunti con altri studenti. I compiti che presenteranno evidenti ed anomale “similitudini” saranno annullati. La prova scritta ha una durata di tre ore, durante le quali non è consentito allontanarsi dall’aula, se non dopo aver consegnato il compito. Si richiede di riportare sul foglio del compito il proprio nominativo completo ed il numero di matricola o un codice identificativo personale equivalente.

Esercizio n. 1

Letto in input un intero $n > 0$, generare un array $W = (w_0, w_1, \dots, w_{n-1})$ di n interi positivi casuali tali che $0 < w_i \leq 30$. Costruire le liste di adiacenza del grafo orientato $G = (V, E)$ con n vertici $V = \{0, 1, \dots, n-1\}$ tale che $(i, j) \in E(G) \iff w_j$ è un multiplo di w_i . Stampare le liste di adiacenza di G e il vettore W .

Esempio Sia $n = 6$ e si consideri il vettore di numeri casuali $W = (w_0 = 12, w_1 = 15, w_2 = 2, w_3 = 24, w_4 = 12, w_5 = 13)$; allora il grafo G prodotto dal programma è rappresentato nella seguente figura.



Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 50
5
6 struct nodo {
7     int info;
8     struct nodo *next;
9 };
10
11 void stampaArray(int A[], int n) {
12     int i;
13     for (i=0; i<n; i++)
14         printf("W[%d] = %d ", i, A[i]);

```

```

15     printf("\n");
16     return;
17 }
18
19 int generaArray(int A[]) {
20     int i, n;
21     srand((unsigned)time(NULL));
22     printf("Numero di elementi: ");
23     scanf("%d", &n);
24     for (i=0; i<n; i++)
25         A[i] = rand() % 30 + 1;
26     return(n);
27 }
28
29 void generaGrafo(struct nodo *G[], int W[], int n) {
30     int i, j;
31     struct nodo *p;
32     for (i=0; i<n; i++)
33         G[i] = NULL;
34     for (i=0; i<n-1; i++) {
35         for (j=0; j<n; j++) {
36             if (i!=j && W[j] % W[i] == 0) {
37                 p = malloc(sizeof(struct nodo));
38                 p->info = j;
39                 p->next = G[i];
40                 G[i] = p;
41             }
42         }
43     }
44     return;
45 }
46
47 void stampaLista(struct nodo *p) {
48     while (p != NULL) {
49         printf("%2d --> ", p->info);
50         p = p->next;
51     }
52     return;
53 }
54
55 void stampaGrafo(struct nodo *G[], int n) {
56     int i;
57     for (i=0; i<n; i++) {
58         printf("%2d: ", i);
59         stampaLista(G[i]);
60         printf("NULL\n");
61     }
62     return;
63 }
64
65 int main(void) {
66     struct nodo *G[MAX];
67     int n, W[MAX];
68     n = generaArray(W);

```

```

69 stampaArray(W, n);
70 generaGrafo(G, W, n);
71 stampaGrafo(G, n);
72 return(0);
73 }

```

Esercizio n. 2

Letto in input un intero $n > 0$ generare un array A di n numeri interi casuali compresi tra 32 e 97 (estremi esclusi). Siano m ed M rispettivamente il valore minimo e il massimo di A . Stampare la più lunga sottosequenza di A con elementi compresi tra m ed M : $m < a_i < M$.

Esempio Sia $n = 8$ e sia $A = (77, 91, 54, 37, 66, 88, 42, 45)$ il vettore di numeri casuali, con valore minimo $m = 37$ e massimo $M = 91$. La più lunga sottosequenza di elementi compresi tra m ed M è 66, 88, 42, 45.

Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 int generaVettore(int A[MAX]) {
7     int i, n;
8     printf("Numero di elementi: ");
9     scanf("%d", &n);
10    srand((unsigned)time(NULL));
11    for (i=0; i<n; i++)
12        A[i] = rand() % 64 + 33;
13    return(n);
14 }
15
16 void stampaVettore(int A[MAX], int n) {
17     int i;
18     for (i=0; i<n; i++)
19         printf("%3d ", A[i]);
20     printf("\n");
21     return;
22 }
23
24 void miniMax(int A[], int n, int *min, int *max) {
25     int i;
26     *min = A[0];
27     *max = A[0];
28     for (i=1; i<n; i++)
29         if (A[i] < *min)
30             *min = A[i];
31         else if (A[i] > *max)
32             *max = A[i];
33     printf("Minimo = %d, massimo = %d\n", *min, *max);
34     return;

```

```

35 }
36
37 void sottosequenza(int A[], int n, int min, int max) {
38     int i, j, cont, contMax=0, start;
39     for (i=0; i<n; i++) {
40         cont = 0;
41         for (j=i; j<n && A[j]>min && A[j]<max; j++)
42             cont++;
43         if (cont > contMax) {
44             contMax = cont;
45             start = i;
46         }
47     }
48     for (i=start; i<start+contMax; i++)
49         printf("%d ", A[i]);
50     return;
51 }
52
53 int main(void) {
54     int A[MAX], n, min, max;
55     n = generaVettore(A);
56     stampaVettore(A, n);
57     miniMax(A, n, &min, &max);
58     sottosequenza(A, n, min, max);
59     return(0);
60 }

```