

Esame scritto del 23 Giugno 2023 (Appello C)

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.

Deve essere consegnata solo la “bella copia” del compito scritto; su ciascun foglio deve essere riportato il **nome**, il **cognome** e il **numero di matricola** (o un altro codice identificativo di fantasia) dello studente.

Esercizio n. 1

Letto in input un intero $n > 0$ costruire un array A con n interi tali che:

$$a_0 = 1$$

$$a_1 = 100$$

$$a_k = \text{numero casuale compreso tra } a_{k-1} \text{ e } a_{k-2} \text{ (estremi inclusi) per } k = 2, \dots, n-1$$

Stampare in output l’array A . Costruire un array B con i soli elementi a_k di A che siano maggiori della media aritmetica degli elementi a_0, \dots, a_k .

Esempio Sia $n = 10$. Supponiamo che l’array A di numeri casuali generato dal programma sia il seguente:

$$A = (1, 100, 23, 58, 46, 58, 50, 52, 51, 52)$$

L’array B è quindi il seguente:

$$B = (100, 46, 58, 50, 52, 51, 52)$$

Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 50
5
6 int min(int x, int y) {
7     int m;
8     if (x < y)
9         m = x;
10    else
11        m = y;
12    return(m);
13 }
14
15 int max(int x, int y) {
16     int m;
17     if (x > y)
18         m = x;
19     else
```

```

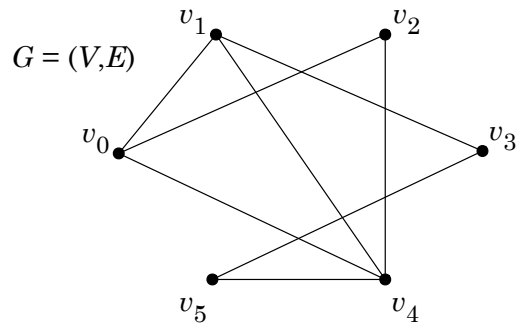
20     m = y;
21     return(m);
22 }
23
24 int generaArray(int X[]) {
25     int i, n;
26     srand((unsigned)time(NULL));
27     printf("Inserisci n: ");
28     scanf("%d", &n);
29     X[0] = 1;
30     X[1] = 100;
31     for (i=2; i<n; i++)
32         X[i] = rand() % (max(X[i-2],X[i-1])-min(X[i-2],X[i-1])+1)+min(X[i-2],X[i-1]));
33     return(n);
34 }
35
36 void stampaArray(int X[], int n) {
37     int i;
38     for (i=0; i<n; i++)
39         printf("%d ", X[i]);
40     printf("\n");
41     return;
42 }
43
44 float media(int X[], int h) {
45     int i, s=0;
46     for (i=0; i<=h; i++)
47         s = s + X[i];
48     return((float)s/(h+1));
49 }
50
51 int main(void) {
52     int A[MAX], B[MAX], n, i, j=0;
53     n = generaArray(A);
54     stampaArray(A, n);
55     for (i = 0; i<n; i++) {
56         if (A[i] > media(A, i)) {
57             B[j] = A[i];
58             j = j+1;
59         }
60     }
61     stampaArray(B, j);
62     return(0);
63 }

```

Esercizio n. 2

Leggere in input le liste di adiacenza di un grafo non orientato $G = (V, E)$ con n vertici v_0, v_1, \dots, v_{n-1} . Visualizzare in output le liste di adiacenza di G . Leggere in input un array C di n interi positivi c_0, c_1, \dots, c_{n-1} . Verificare che per ogni $(u, v) \in E(G)$ risulti $c_u \neq c_v$.

Esempio Si consideri il grafo G con $n = 6$ vertici rappresentato in figura. Sia $C = (1, 3, 2, 1, 1, 3)$. In questo caso la proprietà non è verificata perché ad esempio per lo spigolo (v_0, v_4) risulta $c_0 = 1$ e $c_4 = 1$.



Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #define MAX 50
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 *V[], int n) {
20     for (int i=0; i<n; i++) {
21         printf("%2d: ", i);
22         stampaLista(V[i]);
23     }
24     return;
25 }
26
27 struct nodo *leggiLista(void) {
28     struct nodo *p, *primo = NULL;
29     int i, n;
30     printf("Numero di elementi: ");
31     scanf("%d", &n);
```

```

32 printf("inserisci %d elementi: ", n);
33 for (i=0; i<n; i++) {
34     p = malloc(sizeof(struct nodo));
35     scanf("%d", &p->info);
36     p->next = primo;
37     primo = p;
38 }
39 return(primo);
40 }
41
42 int leggiGrafo(struct nodo *V[]) {
43     int i, n;
44     printf("Numero di vertici: ");
45     scanf("%d", &n);
46     for (i=0; i<n; i++)
47         V[i] = leggiLista();
48     return(n);
49 }
50
51 void leggiArray(int C[], int n) {
52     int i;
53     printf("Inserisci %d interi positivi: ", n);
54     for (i=0; i<n; i++)
55         scanf("%d", &C[i]);
56     return;
57 }
58
59 void stampaArray(int C[], int n) {
60     int i;
61     for (i=0; i<n; i++)
62         printf("%d ", C[i]);
63     printf("\n");
64     return;
65 }
66
67 int verifica(struct nodo *G[], int C[], int n) {
68     int i, flag = 1;
69     struct nodo *p;
70     for (i=0; i<n && flag == 1; i++) {
71         p = G[i];
72         while (p != NULL && C[i] != C[p->info])
73             p = p->next;
74         if (p != NULL) {
75             flag = 0;
76             printf("C[%d] = C[%d] = %d\n", i, p->info, C[i]);
77         }
78     }
79     return(flag);
80 }
81
82 int main(void) {
83     struct nodo *G[MAX];
84     int n, C[MAX];
85     n = leggiGrafo(G);
86     stampaGrafo(G, n);
87     leggiArray(C, n);

```

```
88 stampaArray(C, n);
89 if (verifica(G, C, n) == 1)
90     printf("C[u] != C[v] per ogni spigolo (u,v)\n");
91 else
92     printf("C[u] = C[v] per qualche spigolo (u,v)\n");
93 return(0);
94 }
```