

Esame scritto del 23 Giugno 2017 (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

Letti in input due interi $n, m > 0$, costruire una matrice A di numeri naturali compresi tra 0 e 10, estremi inclusi. Visualizzare in output la matrice. Stampare una delle colonne con il massimo numero di elementi uguali fra loro.

Esempio Supponiamo di aver acquisito in input la seguente matrice 5×6 :

$$A = \begin{pmatrix} 4 & 8 & 3 & 1 & 8 & 2 \\ 2 & 3 & 0 & 2 & 7 & 6 \\ 1 & 6 & 2 & 3 & 8 & 5 \\ 9 & 3 & 3 & 4 & 6 & 9 \\ 1 & 6 & 10 & 5 & 7 & 9 \end{pmatrix}$$

In output viene stampata la seconda colonna, (8, 3, 6, 3, 6), che ha quattro elementi uguali fra loro. Una soluzione ugualmente valida è data dalla visualizzazione della quinta colonna che ha, come la seconda, quattro elementi uguali: (8, 7, 8, 6, 7).

Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 void generaMatrice(int A[MAX][MAX], int n, int m) {
7     int i, j;
8     srand((unsigned)time(NULL));
9     for (i=0; i<n; i++)
10        for (j=0; j<m; j++)
11            A[i][j] = rand() % 11;
12     return;
13 }
14
```

```

15 void stampaMatrice(int A[MAX][MAX], int n, int m) {
16     int i, j;
17     for (i=0; i<n; i++) {
18         for (j=0; j<m; j++)
19             printf("%2d ", A[i][j]);
20         printf("\n");
21     }
22     return;
23 }
24
25 int maxDoppioni(int A[MAX][MAX], int n, int m) {
26     int i, j, k, c=0, s, smax=0;
27     for (j=0; j<m; j++) {
28         s = 0;
29         for (i=0; i<n-1; i++)
30             for (k=i+1; k<n; k++)
31                 if (A[i][j] == A[k][j])
32                     s++;
33         if (s > smax) {
34             c = j;
35             smax = s;
36         }
37     }
38     return(c);
39 }
40
41 int main(void) {
42     int A[MAX][MAX], n, m, i, c;
43     printf("Numero di righe e di colonne: ");
44     scanf("%d %d", &n, &m);
45     generaMatrice(A, n, m);
46     stampaMatrice(A, n, m);
47     c = maxDoppioni(A, n, m);
48     for(i=0; i<n; i++)
49         printf("%d ", A[i][c]);
50     printf("\n\n");
51     return(0);
52 }

```

Esercizio n. 2

Letta in input una sequenza di numeri interi, memorizzarla in una lista L e visualizzare in output la lista. Eliminare dalla lista tutti gli elementi privi di "doppioni". Stampare la nuova lista.

Esempio Supponiamo di aver acquisito in input la seguente lista:

$L : 6 \rightarrow 4 \rightarrow 7 \rightarrow 7 \rightarrow 3 \rightarrow 9 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow \text{null}$

La lista ottenuta dopo l'eliminazione degli elementi privi di doppioni è la seguente:

$L : 4 \rightarrow 7 \rightarrow 7 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 5 \rightarrow 2 \rightarrow 5 \rightarrow \text{null}$

Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 struct nodo {
5     int info;
6     struct nodo *next;
7 };
8
9 struct nodo *leggiLista(void) {
10     int n, i;
11     struct nodo *p, *primo = NULL;
12     printf("Numero di elementi: ");
13     scanf("%d", &n);
14     printf("Inserisci %d elementi della lista: ", n);
15     for (i=0; i<n; i++) {
16         p = malloc(sizeof(struct nodo));
17         scanf("%d", &p->info);
18         p->next = primo;
19         primo = p;
20     }
21     return(primo);
22 }
23
24 void stampaLista(struct nodo *p) {
25     while (p!=NULL) {
26         printf("%d --> ", p->info);
27         p = p->next;
28     }
29     printf("NULL\n");
30     return;
31 }
32
```

```

33 struct nodo *eliminaSingoli(struct nodo *L) {
34     struct nodo *p, *q, *prec = NULL;
35     int flag;
36     p = L;
37     while (p != NULL) {
38         flag = 0;
39         q = L;
40         while (q != p && q->info != p->info)
41             q = q->next;
42         if (q != p) {
43             flag = 1;
44         } else {
45             q = p->next;
46             while (q != NULL && q->info != p->info)
47                 q = q->next;
48             if (q != NULL)
49                 flag = 1;
50         }
51         if (flag == 0) {
52             if (p == L) {
53                 L = L->next;
54                 free(p);
55                 p = L;
56             } else {
57                 prec->next = p->next;
58                 free(p);
59                 p = prec->next;
60             }
61         } else {
62             prec = p;
63             p = p->next;
64         }
65     }
66     return(L);
67 }
68
69 int main(void) {
70     struct nodo *L;
71     L = leggiLista();
72     stampaLista(L);
73     L = eliminaSingoli(L);
74     stampaLista(L);
75     return(0);
76 }

```