

Esame scritto del 8 Settembre 2023 (Appello X)

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 tre numeri interi positivi, n , m e k , generare una matrice A con n righe e m colonne composta da numeri interi casuali compresi nell’intervallo $[-k, k]$. Stampare la matrice. Stampare in output la colonna la cui somma algebrica degli elementi è più vicina a zero. Ricordiamo che il valore assoluto di un numero intero x si calcola con la funzione $\text{abs}(x)$.

Esempio Siano $n = 5$, $m = 6$ e $k = 10$ e supponiamo sia stata generata la seguente matrice A di numeri casuali con 5 righe, 6 colonne e valori compresi tra -10 e 10 :

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

La colonna la cui somma degli elementi è più vicina a zero è la quinta, infatti risulta $5 - 9 + 10 + 1 - 5 = 2$, mentre le altre colonne hanno una somma maggiore o uguale a 2 in valore assoluto.

Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 50
5
6 void generaMatrice(int A[MAX][MAX], int n, int m, int k) {
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() % (2*k+1) - k;
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("%3d ", A[i][j]);

```

```

20     printf("\n");
21 }
22 return;
23 }
24
25 int calcolaColonna(int A[MAX][MAX], int n, int m) {
26     int i, j, s, smin = 0, jmin = 0;
27     for (i=0; i<n; i++)
28         smin = smin + A[i][0];
29     for (j=1; j<m; j++) {
30         s = 0;
31         for (i=0; i<n; i++)
32             s = s + A[i][j];
33         if (abs(s) < abs(smin)) {
34             smin = s;
35             jmin = j;
36         }
37     }
38     return jmin;
39 }
40
41 void stampaColonna(int A[MAX][MAX], int n, int j) {
42     int i;
43     printf("Soluzione: ");
44     for (i=0; i<n; i++)
45         printf("%d ", A[i][j]);
46     printf("\n");
47     return;
48 }
49
50 int main(void) {
51     int A[MAX][MAX], n, m, h, k;
52     printf("Inserisci tre interi positivi: ");
53     scanf("%d %d %d", &n, &m, &k);
54     generaMatrice(A, n, m, k);
55     stampaMatrice(A, n, m);
56     h = calcolaColonna(A, n, m);
57     stampaColonna(A, n, h);
58     return 0;
59 }

```

Esercizio n. 2

Letti in input due interi positivi n e k , generare una lista L_1 di n interi positivi casuali minori di k . Costruire una seconda lista L_2 di n elementi tale che l'elemento i -esimo contenga il numero di elementi della prima lista minori dell' i -esimo elemento della prima lista. Stampare in output le due liste.

Esempio Sia $n = 10$ e $k = 20$; supponiamo che la lista L_1 di numeri casuali minori di k sia la seguente:

$$L_1 = 12 \rightarrow 2 \rightarrow 13 \rightarrow 16 \rightarrow 10 \rightarrow 5 \rightarrow 15 \rightarrow 6 \rightarrow 10 \rightarrow 8$$

Allora la lista L_2 è la seguente:

$$L_2 = 6 \rightarrow 0 \rightarrow 7 \rightarrow 9 \rightarrow 4 \rightarrow 1 \rightarrow 8 \rightarrow 2 \rightarrow 4 \rightarrow 3$$

Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4
5 struct nodo {
6     int info;
7     struct nodo *next;
8 };
9
10 struct nodo *costruisciLista(struct nodo *L1) {
11     struct nodo *p, *q, *r, *L2 = NULL;
12     p = L1;
13     while (p != NULL) {
14         q = malloc(sizeof(struct nodo));
15         q->info = 0;
16         q->next = L2;
17         L2 = q;
18         p = p->next;
19     }
20     p = L1;
21     q = L2;
22     while (p != NULL) {
23         r = L1;
24         while (r != NULL) {
25             if (r->info < p->info)
26                 q->info++;
27             r = r->next;
28         }
29         p = p->next;
30         q = q->next;
31     }
32     return L2;
33 }
34
35 void stampaLista(struct nodo *p) {
36     while (p != NULL) {
37         printf("%d --> ", p->info);
38         p = p->next;
39     }
40     printf("null\n");
```

```

41     return;
42 }
43
44 struct nodo *generaLista(int n, int k) {
45     struct nodo *p, *primo = NULL;
46     int i;
47     srand((unsigned)time(NULL));
48     for (i=0; i<n; i++) {
49         p = malloc(sizeof(struct nodo));
50         p->info = rand() % k;
51         p->next = primo;
52         primo = p;
53     }
54     return primo;
55 }
56
57 int main(void) {
58     struct nodo *L1, *L2;
59     int n, k;
60     printf("Inserisci due numeri interi positivi: ");
61     scanf("%d %d", &n, &k);
62     L1 = generaLista(n, k);
63     stampaLista(L1);
64     L2 = costruisciLista(L1);
65     stampaLista(L2);
66     return 0;
67 }

```