

Esame scritto del 8 Settembre 2017 (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

Letta in input una sequenza di n numeri interi, memorizzarla in una lista L . Verificare se gli elementi di L sono disposti alternando ad un elemento più grande del successivo un elemento più piccolo. Il primo elemento della lista può essere più piccolo o più grande del successivo, indifferentemente. Stampare “Si” se la lista rispetta la regola dell’alternanza tra elementi piccoli e grandi, stampare “No” altrimenti.

Esempio Si considerino le seguenti liste: entrambe rispettano la regola dell’alternanza:

$L : 7 \rightarrow 2 \rightarrow 18 \rightarrow 5 \rightarrow 11 \rightarrow 3 \rightarrow null$

$L : 6 \rightarrow 15 \rightarrow 14 \rightarrow 32 \rightarrow 1 \rightarrow 4 \rightarrow null$

La seguente lista invece non rispetta la regola dell’alternanza:

$L : 3 \rightarrow 12 \rightarrow 7 \rightarrow 5 \rightarrow 8 \rightarrow 2 \rightarrow null$

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 i, n;
11     struct nodo *p, *primo=NULL;
12     printf("Numero di elementi: ");
13     scanf("%d", &n);
14     printf("Inserisci i %d elementi della lista: ", n);
15     for (i=0; i<n; i++) {
```

```

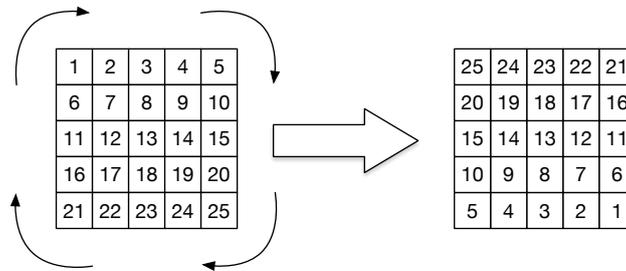
16     p = malloc(sizeof(struct nodo));
17     scanf("%d", &p->info);
18     p->next = primo;
19     primo = p;
20 }
21 return(primo);
22 }
23
24 int verificaAlternanza(struct nodo *L) {
25     int flag=1;
26     if (L != NULL && L->next != NULL) {
27         if (L->info > L->next->info)
28             flag = 1;
29         else if (L->info < L->next->info)
30             flag = -1;
31         else
32             flag = 0;
33         L = L->next;
34         while (flag != 0 && L->next != NULL) {
35             if (flag == 1 && L->info < L->next->info)
36                 flag = -1;
37             else if (flag == -1 && L->info > L->next->info)
38                 flag = 1;
39             else
40                 flag = 0;
41             L = L->next;
42         }
43     }
44     return(flag*flag);
45 }
46
47 int main(void) {
48     struct nodo *L;
49     L = leggiLista();
50     if (verificaAlternanza(L) == 1)
51         printf("Si'\n");
52     else
53         printf("No\n");
54     return(0);
55 }

```

Esercizio n. 2

Letto in input un numero intero $n > 0$ generare una matrice quadrata A di ordine n con numeri interi casuali compresi nell'intervallo $[0;99]$; stampare la matrice A . Spostare gli elementi della matrice ruotandola di 180° e stampare la nuova matrice.

Esempio



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() % 100;
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 void scambia(int *x, int *y) {
26     int z;
27     z = *x;
28     *x = *y;
```

```

29     *y = z;
30     return;
31 }
32
33 void scambiaRighe(int A[MAX][MAX], int n) {
34     int i, j;
35     for (i=0; i<n/2; i++)
36         for (j=0; j<n; j++)
37             scambia(&A[i][j], &A[n-1-i][j]);
38     return;
39 }
40
41 void scambiaColonne(int A[MAX][MAX], int n) {
42     int i, j;
43     for (j=0; j<n/2; j++)
44         for (i=0; i<n; i++)
45             scambia(&A[i][j], &A[i][n-1-j]);
46     return;
47 }
48
49 int main(void) {
50     int A[MAX][MAX], n;
51     printf("Ordine della matrice quadrata (numero righe e colonne): ");
52     scanf("%d", &n);
53     generaMatrice(A, n, n);
54     printf("Matrice casuale:\n");
55     stampaMatrice(A, n, n);
56     scambiaRighe(A, n);
57     scambiaColonne(A, n);
58     printf("\nMatrice ruotata di 180 gradi:\n");
59     stampaMatrice(A, n, n);
60     return(0);
61 }

```