

## Esame scritto del 10 settembre 2010

*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

Letti in input due interi  $n, k > 0$  generare una sequenza di  $n$  interi casuali compresi tra  $-k$  e  $2k$ , estremi inclusi, e memorizzarli in una lista. Stampare la lista. Modificare la lista inserendo un elemento in mezzo ad ogni coppia di elementi preesistenti nella lista originale, il cui valore sia la media aritmetica degli altri due. Stampare la lista modificata in questo modo.

**Esempio** Siano  $n = 10$  e  $k = 5$ . Se la lista di numeri casuali originale è la seguente:

$4 \rightarrow -3 \rightarrow 9 \rightarrow 7 \rightarrow 0 \rightarrow -1 \rightarrow 6 \rightarrow 10 \rightarrow 9 \rightarrow -2$

la lista prodotta dal programma è la seguente (in grassetto gli elementi inseriti fra le coppie di elementi della lista originale):

$4 \rightarrow \mathbf{0,5} \rightarrow -3 \rightarrow \mathbf{3} \rightarrow 9 \rightarrow \mathbf{8} \rightarrow 7 \rightarrow \mathbf{3,5} \rightarrow 0 \rightarrow \mathbf{-0,5} \rightarrow -1 \rightarrow \mathbf{2,5} \rightarrow 6 \rightarrow \mathbf{8} \rightarrow 10 \rightarrow \mathbf{9,5} \rightarrow 9 \rightarrow \mathbf{3,5} \rightarrow -2$

### Soluzione

```

1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <time.h>
4  #define MAX 100
5
6  struct nodo {
7      float info;
8      struct nodo *next;
9  };
10
11 struct nodo *costruisciLista() {
12     int n, i, k;
13     struct nodo *p, *primo=NULL;
14     printf("Numero di elementi: ");
15     scanf("%d", &n);
16     printf("Valore soglia: ");
17     scanf("%d", &k);
18     srand((unsigned)time(NULL));
19     for (i=0; i<n; i++) {
20         p = malloc(sizeof(struct nodo));
21         p->next = primo;
22         primo = p;

```

```

23     p->info = rand() % (3*k+1) - k;
24 }
25 return(primo);
26 }
27
28 void modificaLista(struct nodo *p) {
29     struct nodo *q;
30     while (p->next != NULL) {
31         q = malloc(sizeof(struct nodo));
32         q->info = (p->info + p->next->info)/2.0;
33         q->next = p->next;
34         p->next = q;
35         p = q->next;
36     }
37     return;
38 }
39
40 void stampaLista(struct nodo *p) {
41     while (p != NULL) {
42         printf("%.2f --> ", p->info);
43         p = p->next;
44     }
45     printf("NULL\n");
46     return;
47 }
48
49 int main(void) {
50     struct nodo *p;
51     p = costruisciLista();
52     stampaLista(p);
53     modificaLista(p);
54     stampaLista(p);
55     return(0);
56 }

```

## Esercizio n. 2

In una matrice con 3 colonne ed  $n$  righe sono raccolti i risultati di un campionato a squadre: su ciascuna riga è riportato il risultato di una partita, indicando nella prima colonna il codice numerico progressivo della squadra (da 1 al numero di squadre) che ha giocato in casa, nella seconda colonna il codice della squadra che ha giocato fuori casa e nella terza il risultato (0 =pareggio, 1 =vittoria della squadra di casa, 2 =vittoria della squadra ospite).

Tenendo conto che una vittoria vale 2 punti, un pareggio 1 punto e una sconfitta 0 punti, calcolare il codice di una delle squadre al vertice della classifica, dopo aver acquisito in input il numero  $n > 0$  e la matrice dei risultati.

**Esempio** Sia  $n = 9$  e si consideri la seguente matrice dei risultati:

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

La squadra al vertice della classifica è la n.6 con 5 punti (2 vittorie, 1 pareggio, 0 sconfitte).

## Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 int leggiMatrice(int A[MAX][3], int *s) {
7     int i, n;
8     printf("Numero di partite disputate: ");
9     scanf("%d", &n);
10    printf("Numero di squadre: ");
11    scanf("%d", s);
12    printf("Inserisci i codici delle squadre e l'esito dell'incontro:\n");
13    for (i=0; i<n; i++)
14        scanf("%d %d %d", &A[i][0], &A[i][1], &A[i][2]);
15    return(n);
16 }
17
18 void calcolaVincitore(int A[MAX][3], int n, int s) {
19     int punti[MAX], i, max, maxs, mins;
20     maxs = 0;
21     mins = MAX;
22     for(i=1; i<=s; i++)
23         punti[i] = 0;
24     for (i=0; i<n; i++) {
25         if (A[i][2] == 0) {
26             punti[A[i][0]]++;
27             punti[A[i][1]]++;
28         } else if (A[i][2] == 1) {
29             punti[A[i][0]]+=2;
30         } else {
31             punti[A[i][1]]+=2;
32         }
33     }
34     max = 1;
35     for (i=2; i<=s; i++) {
36         if (punti[max] < punti[i])
37             max = i;
38     }
```

```
39 | printf("La squadra %d e' in testa con %d punti.\n", max, punti[max]);
40 | return;
41 | }
42 |
43 | void stampaMatrice(int A[MAX][3], int n) {
44 |     int i, j;
45 |     for (i=0; i<n; i++) {
46 |         for (j=0; j<3; j++)
47 |             printf("%d ", A[i][j]);
48 |         printf("\n");
49 |     }
50 |     return;
51 | }
52 |
53 | int main(void) {
54 |     int n, s, A[MAX][3];
55 |     n = leggiMatrice(A, &s);
56 |     stampaMatrice(A, n);
57 |     calcolaVincitore(A, n, s);
58 |     return(0);
59 | }
```