

Corso di Informatica 1 (IN110) – Prof. Marco Liverani – a.a. 2014/2015

Esame scritto del 23 Gennaio 2015 (Appello A)

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 $n, m > 0$, costruire una matrice A di ordine $n \times m$ (n righe ed m colonne) di numeri interi casuali compresi tra -37 e 59 , estremi inclusi. Costruire una matrice B di ordine $k \times n$ (k righe e n colonne, con $k \leq m$) che abbia come righe le colonne di A con almeno la metà degli elementi maggiori di zero. Stampare la matrice A e la matrice B .

Esempio Sia $n = 4$ e $m = 6$; consideriamo la seguente matrice:

$$A = \begin{pmatrix} 12 & -5 & 12 & -6 & 9 \\ 4 & -2 & -14 & 59 & -10 \\ -6 & 36 & 12 & -37 & -6 \\ 8 & -11 & -9 & 3 & -10 \end{pmatrix}$$

Le colonne di A con almeno $n/2 = 2$ elementi positivi sono la prima, la terza e la quarta; quindi la matrice B in questo caso è la seguente matrice di $k = 3$ righe e $n = 4$ colonne:

$$B = \begin{pmatrix} 12 & 4 & -6 & 8 \\ 12 & -14 & 12 & -9 \\ -6 & 59 & -37 & 3 \end{pmatrix}$$

Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 100
5
6 void genera(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() % 97 - 37;
12     return;
13 }
14
15 int costruisci(int B[MAX][MAX], int A[MAX][MAX], int n, int m) {
16     int k=0, i, j, c;
17     for (j=0; j<m; j++) {
18         c = 0;
19         for (i=0; i<n; i++)
20             if (A[i][j] > 0)
21                 c++;
22         if (c >= (float)n/2) {
23             for (i=0; i<n; i++)
24                 B[k][i] = A[i][j];
25             k++;
26         }
27     }
28     return(k);
29 }
30
31 void stampaMatrice(int X[MAX][MAX], int n, int m) {
32     int i, j;
33     for (i=0; i<n; i++) {
34         for (j=0; j<m; j++)
35             printf("%3d ", X[i][j]);
36         printf("\n");
37     }
38     printf("\n");
39     return;
40 }
41
42 int main(void) {
43     int A[MAX][MAX], B[MAX][MAX], n, m, k;
44     printf("Numero di righe e di colonne: ");
45     scanf("%d %d", &n, &m);
46     genera(A, n, m);
47     k = costruisci(B, A, n, m);
48     stampaMatrice(A, n, m);
49     stampaMatrice(B, k, n);
50     return(0);
51 }

```

Esercizio n. 2

Leggere in input una sequenza di $n > 0$ numeri naturali e memorizzarli in una lista. Costruire una seconda lista con gli elementi della prima e la frequenza rapportata al numero complessivo di elementi di ciascun elemento. Stampare la seconda lista.

Esempio Sia $n = 7$ e consideriamo la seguente lista acquisita in input:

$7 \rightarrow 5 \rightarrow 12 \rightarrow 5 \rightarrow 13 \rightarrow 7 \rightarrow 5$

La lista con le frequenze degli elementi della prima è la seguente:

$(7, 0.29) \rightarrow (5, 0.43) \rightarrow (12, 0.14) \rightarrow (5, 0.43) \rightarrow (13, 0.14) \rightarrow (7, 0.29) \rightarrow (5, 0.43)$

Soluzione

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

```

32     printf("%d, %1.2f) --> ", p->info, p->frequenza);
33     p = p->next;
34 }
35 printf("NULL\n");
36 return;
37 }
38
39 int conta(struct nodo *p, int x) {
40     int c=0;
41     while (p != NULL) {
42         if (p->info == x)
43             c++;
44         p = p->next;
45     }
46     return(c);
47 }
48
49 int numeroElementi(struct nodo *p) {
50     int c=0;
51     while (p != NULL) {
52         c++;
53         p = p->next;
54     }
55     return(c);
56 }
57
58 struct nodoBis *costruisci(struct nodo *p) {
59     struct nodoBis *q, *q1 = NULL;
60     struct nodo *p1;
61     int n;
62     p1 = p;
63     n = numeroElementi(p1);
64     while (p != NULL) {
65         q = malloc(sizeof(struct nodoBis));
66         q->info = p->info;
67         q->frequenza = (float)conta(p1, p->info)/n;
68         q->next = q1;
69         q1 = q;
70         p = p->next;
71     }
72     return(q1);
73 }
74
75 int main(void) {
76     struct nodo *primo;
77     struct nodoBis *primoBis;
78     primo = leggiLista();

```

```
79 | primoBis = costruisci(primo);  
80 | stampaLista(primoBis);  
81 | return(0);  
82 | }
```