

Corso di Informatica 1 (IN110) – Prof. Marco Liverani – a.a. 2021/2022

Esame scritto del 17 Giugno 2022 (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 positivi n e m , con $n < m$, generare due array A e B rispettivamente di n ed m numeri casuali in $\{0, 1\}$. Visualizzare in output A e B . Verificare se A è un sotto-array di B e, in caso positivo, calcolare quante volte A è contenuto in B .

Esempio Siano $n = 4$ e $m = 15$ e si considerino i seguenti array di numeri casuali in $\{0, 1\}$:

$$\begin{aligned}
 A &= (0, 0, 1, 0, 1) \\
 B &= (1, \underbrace{0, 0, 1, 0, 1}_A, 0, \underbrace{0, 0, 1, 0, 1}_A, 1, 0, 1)
 \end{aligned}$$

Risulta che A è contenuto in B due volte.

Soluzione

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4
5 #define MAX 50
6
7 int generaArrayBinario(int X[]) {
8     int i, n;
9     printf("Numero di elementi: ");
10    scanf("%d", &n);
11    for (i=0; i<n; i++)
12        X[i] = rand() % 2;
13    return(n);
14 }
15
16 void stampaArray(int X[], int n) {
17     int i;
18     for (i=0; i<n; i++)
19         printf("%d ", X[i]);
20     printf("\n");
21     return;
22 }
23
24

```

```

25 int sottoArray(int A[], int n, int B[], int m) {
26     int k = 0, i, j;
27     for (i=0; i<m-n+1; i++) {
28         j = 0;
29         while (j < n && A[j] == B[i+j])
30             j++;
31         if (j == n)
32             k++;
33     }
34     return(k);
35 }
36
37 int main(void) {
38     int A[MAX], B[MAX], n, m, k;
39     srand((unsigned)time(NULL));
40     n = generaArrayBinario(A);
41     m = generaArrayBinario(B);
42     k = sottoArray(A, n, B, m);
43     stampaArray(A, n);
44     stampaArray(B, m);
45     printf("Il primo array e' contenuto nel secondo %d volte\n", k);
46     return(0);
47 }

```

Esercizio n. 2

Letto in input un intero $n > 0$, generare una lista L_1 contenente n numeri interi casuali compresi tra 1 e 100, estremi inclusi. Costruire una seconda lista L_2 con tutti gli elementi dispari di L_1 disposti in un ordine qualsiasi, seguiti poi da tutti gli elementi pari. Visualizzare in output L_1 e L_2 .

Esempio Si consideri la seguente lista L_1 di 15 numeri interi casuali compresi tra 1 e 100:

$L_1 : 31 \rightarrow 36 \rightarrow 79 \rightarrow 36 \rightarrow 96 \rightarrow 62 \rightarrow 80 \rightarrow 89 \rightarrow 13 \rightarrow 94 \rightarrow 93 \rightarrow 29 \rightarrow 93 \rightarrow 24 \rightarrow 71$

La lista L_2 prodotta dal programma è la seguente:

$L_2 = \underbrace{71 \rightarrow 93 \rightarrow 29 \rightarrow 93 \rightarrow 13 \rightarrow 89 \rightarrow 79 \rightarrow 31}_{\text{dispari}} \rightarrow \underbrace{24 \rightarrow 94 \rightarrow 80 \rightarrow 62 \rightarrow 96 \rightarrow 36 \rightarrow 36}_{\text{pari}}$

Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #define MAX 50
5
6 struct nodo {
7     int info;
8     struct nodo *next;
9 };
10
11 struct nodo *listaPariDispari(struct nodo *L1) {
12     struct nodo *P = NULL, *D = NULL, *q, *r, *u;
13     r = L1;
14     while (r != NULL) {
15         q = malloc(sizeof(struct nodo));
16         q->info = r->info;
17         if (r->info % 2 != 0) {
18             if (D == NULL)
19                 u = q;
20             q->next = D;
21             D = q;
22         } else {
23             q->next = P;
24             P = q;
25         }
26         r = r->next;
27     }
28     if (D != NULL) {
29         u->next = P;
30         P = D;
31     }
32     return(P);
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 *listaRandom(void) {
45     struct nodo *p, *primo=NULL;
46     int i, n;
47     printf("Numero di elementi: ");
48     scanf("%d", &n);
49     for (i=0; i<n; i++) {
50         p = malloc(sizeof(struct nodo));
51         p->info = rand() % 100 + 1;
52         p->next = primo;
53         primo = p;
54     }
55     return(primo);
56 }
57
58 int main(void) {
59     struct nodo *L1, *L2;
60     srand((unsigned)time(NULL));
61     L1 = listaRandom();
62     L2 = listaPariDispari(L1);
63     stampaLista(L1);
64     stampaLista(L2);
65     return(0);
66 }

```