

Corso di Algoritmi e Strutture Dati (IN110)

Tutorato n. 8

Marco Liverani*

Esercizio n. 1

Letta in input una sequenza di numeri interi positivi memorizzarla in una lista. Costruire una seconda lista contenente soltanto gli elementi della prima lista che non siano numeri primi. Stampare la seconda lista.

Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <math.h>
4
5 struct nodo {
6     int info;
7     struct nodo *next;
8 };
9
10 int primo(int x) {
11     int i, flag;
12     flag = 1;
13     i = 2;
14     while (i<=sqrt(x) && flag==1) {
15         if (x%i == 0)
16             flag = 0;
17         i = i+1;
18     }
19     return(flag);
20 }
21
22 struct nodo *leggi_lista(void) {
23     struct nodo *p, *primo;
24     int i, n;
25     primo = NULL;
26     printf("Numero di elementi: ");
27     scanf("%d", &n);
28     printf("inserisci %d numeri interi positivi: ", n);
29     for (i=0; i<n; i++) {
```

*Università degli Studi Roma Tre, Corso di Laurea in Matematica, Corso di Algoritmi e Strutture Dati (IN110); e-mail liverani@mat.uniroma3.it - sito web del corso <http://www.mat.uniroma3.it/users/liverani/IN110/>

```

30     p = malloc(sizeof(struct nodo));
31     scanf("%d", &p->info);
32     p->next = primo;
33     primo = p;
34 }
35 return(primo);
36 }
37
38 void stampa_lista(struct nodo *p) {
39     while (p != NULL) {
40         printf("%d ", p->info);
41         p = p->next;
42     }
43     printf("\n");
44     return;
45 }
46
47 int main(void) {
48     struct nodo *p, *q, *r;
49     p = leggi_lista();
50     r = NULL;
51     while (p != NULL) {
52         if (primo(p->info) == 0) {
53             q = malloc(sizeof(struct nodo));
54             q->info = p->info;
55             q->next = r;
56             r = q;
57         }
58         p = p->next;
59     }
60     stampa_lista(r);
61     return(0);
62 }

```

Esercizio n. 2

Letta in input una lista, costruire una seconda lista con gli elementi disposti al contrario rispetto alla lista originale (es.: se la lista letta in input è $1 \rightarrow 8 \rightarrow 4$ viene costruita la lista $4 \rightarrow 8 \rightarrow 1$). Stampare la seconda lista.

Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 struct nodo {
5     int info;
6     struct nodo *next;
7 };
8
9 struct nodo *leggi_lista(void) {
10     struct nodo *p, *primo=NULL;
11     int i, n;
12     printf("Numero di elementi: ");
13     scanf("%d", &n);
14     printf("Inserisci %d numeri: ", 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 void stampa_lista(struct nodo *p) {
25     while (p != NULL) {
26         printf("%d --> ", p->info);
27         p = p->next;
28     }
29     printf("NULL\n");
30 }
31
32 struct nodo *inverti(struct nodo *p) {
33     struct nodo *q, *q1 = NULL;
34     while (p != NULL) {
35         q = malloc(sizeof(struct nodo));
36         q->info = p->info;
37         q->next = q1;
38         q1 = q;
39         p = p->next;
40     }
41     return(q1);
42 }
43
44 int main(void) {
45     struct nodo *p, *q;
46     p = leggi_lista();
```

```
47 | q = inverti(p);  
48 | stampa_lista(q);  
49 | return(0);  
50 | }
```

Esercizio n. 3

Lette in input due liste di numeri interi ognuna delle quali ordinata, costruire una terza lista di numeri interi ordinata, mediante la “fusione” delle prime due. Stampare la lista.

Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 struct nodo {
5     int info;
6     struct nodo *next;
7 };
8
9 struct nodo *fusione(struct nodo *p1, struct nodo *p2) {
10     struct nodo *p, *primo = NULL;
11     while (p1 != NULL && p2 != NULL) {
12         p = malloc(sizeof(struct nodo));
13         p->next = primo;
14         primo = p;
15         if (p1->info > p2->info) {
16             p->info = p1->info;
17             p1 = p1->next;
18         } else {
19             p->info = p2->info;
20             p2 = p2->next;
21         }
22     }
23     while (p1 != NULL) {
24         p = malloc(sizeof(struct nodo));
25         p->info = p1->info;
26         p->next = primo;
27         primo = p;
28         p1 = p1->next;
29     }
30     while (p2 != NULL) {
31         p = malloc(sizeof(struct nodo));
32         p->info = p2->info;
33         p->next = primo;
34         primo = p;
35         p2 = p2->next;
36     }
37     return(primo);
38 }
39
40 void stampa_lista(struct nodo *p) {
41     while (p != NULL) {
42         printf("%d --> ", p->info);
43         p = p->next;
44     }
45     printf("NULL\n");
46 }
47
```

```

48 struct nodo *leggi_lista(void) {
49     struct nodo *p, *primo=NULL;
50     int i, n;
51     printf("Numero di elementi: ");
52     scanf("%d", &n);
53     printf("Inserisci %d numeri: ", n);
54     for (i=0; i<n; i++) {
55         p = malloc(sizeof(struct nodo));
56         scanf("%d", &p->info);
57         p->next = primo;
58         primo = p;
59     }
60     return(primo);
61 }
62
63 int main(void) {
64     struct nodo *p, *q, *r;
65     p = leggi_lista();
66     q = leggi_lista();
67     r = fusione(p, q);
68     stampa_lista(r);
69     return(0);
70 }

```

Esercizio n. 4

Letta in input una lista di numeri interi stamparla e poi ordinarla utilizzando l'algoritmo *selection sort*. Stampare la lista ordinata.

Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 struct nodo {
5     int info;
6     struct nodo *next;
7 };
8
9 void scambia(int *a, int *b) {
10     int c;
11     c = *a;
12     *a = *b;
13     *b = c;
14     return;
15 }
16
17 void selection_sort(struct nodo *p) {
18     struct nodo *q;
19     while (p != NULL) {
20         q = p->next;
21         while (q != NULL) {
22             if (p->info > q->info)
23                 scambia(&p->info, &q->info);
24             q = q->next;
25         }
26         p = p->next;
27     }
28     return;
29 }
30
31 struct nodo *leggi_lista(void) {
32     struct nodo *p, *primo=NULL;
33     int i, n;
34     printf("Numero di elementi: ");
35     scanf("%d", &n);
36     printf("Inserisci %d numeri: ", n);
37     for (i=0; i<n; i++) {
38         p = malloc(sizeof(struct nodo));
39         scanf("%d", &p->info);
40         p->next = primo;
41         primo = p;
42     }
43     return(primo);
44 }
45
46 void stampa_lista(struct nodo *p) {
47     while (p != NULL) {
```

```
48     printf("%d --> ", p->info);
49     p = p->next;
50 }
51 printf("NULL\n");
52 }
53
54 int main(void) {
55     struct nodo *p;
56     p = leggi_lista();
57     stampa_lista(p);
58     selection_sort(p);
59     stampa_lista(p);
60     return(1);
61 }
```