

# Laboratorio di Programmazione: Linguaggio C

## Lezione 22 del 26 maggio 2014

Damiano Macedonio

### Esercizio 1 [Tratto dall'esame del 7 febbraio 2014]

Si scriva una funzione

```
int included(const char *where, const char *what)
```

che determina se i caratteri di `what` si trovano tutti dentro `where` e nello stesso ordine. Per esempio, l'esecuzione del seguente programma:

```
1 #include <stdio.h>
2
3 int main(void) {
4     const char *s = "Major Tom to Ground Control";
5
6     printf("%i\n", included(s, "jomo"));
7     printf("%i\n", included(s, "Tootro"));
8     printf("%i\n", included(s, "troll"));
9     printf("%i\n", included(s, ""));
10    printf("%i\n", included("", "troll"));
11
12    return 0;
13 }
```

dovrà stampare:

```
1
1
0
1
0
```

## Esercizio 2 [Tratto dalla prova parziale del 12 giugno 2013]

Si consideri la struttura *lista di caratteri* definita come segue:

```
struct list {  
    char head;  
    struct list *tail;  
};
```

e si utilizzi il puntatore NULL per determinare la fine della lista.

Si definiscano le funzioni:

```
struct list *construct_list_from_string(const char *s)
```

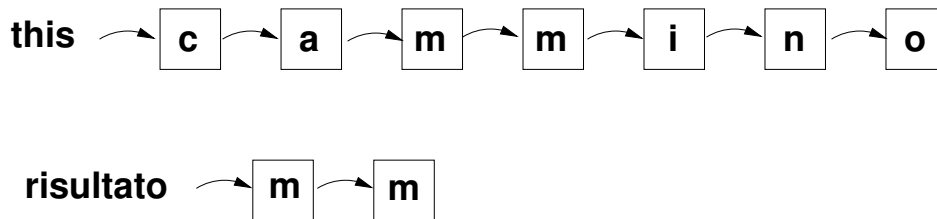
che restituisce una lista contenente i caratteri di *s*, nello stesso ordine;

```
void print_list(struct list l)
```

che stampa a video la lista *l* di caratteri;

```
struct list *doppie(struct list *this)
```

che riceve una lista di caratteri *this*, con al massimo due caratteri uguali di seguito, e restituisce una lista di caratteri fatta dalle doppie contenute in *this*. In altri termini, nel risultato ci sono solo i caratteri che sono uguali al precedente o al successivo, come nel seguente esempio:



Se tutto è corretto, un'esecuzione del seguente programma:

```
1 int main(void) {
2     char buffer[100];
3     struct list *l;
4
5     printf("Inserisci una frase: ");
6     scanf("%s", buffer);
7
8     if (al_massimo_due_di_seguito(buffer)) { // si veda lezione_21
9         printf("Non ci sono piu' di due lettere uguali di seguito\n");
10        l = construct_list_from_string(buffer);
11        print_list(l);
12        printf("\n");
13        l = doppie(l);
14        print_list(l);
15        printf("\n");
16    }
17    else
18        printf("Ci sono piu' di due lettere uguali di seguito\n");
19
20    return 0;
21 }
```

è la seguente:

```
Inserisci una frase: ammettere
Non ci sono piu' di due lettere uguali di seguito
[a, m, m, e, t, t, e, r, e]
[m, m, t, t]
```

Un'altra esecuzione è:

```
Inserisci una frase: cammmmino
Ci sono piu' di due lettere uguali di seguito
```

### Esercizio 3 [Tratto dalla prova parziale del 7 febbraio 2013]

Si scrivano i file `chart.h` e `chart.c` che definiscono e implementano una classifica delle dieci canzoni più ascoltate. Devono essere realizzate le seguenti funzioni:

```
struct chart *construct_chart();
void destroy_chart(struct chart *this);
void print_chart(struct chart *this);
void set_song_title(struct chart *this, const char *title, int position);
```

L'ultima funzione assegna il titolo della canzone posizionata tra 1 e 10 inclusi. La terza stampa la classifica, numerando le canzoni da 1 a 10, usando un trattino per le posizioni per cui non si è ancora assegnata la canzone. Quando una classifica è stata appena costruita, nessuna posizione ha ancora una canzone assegnata.

Se tutto è corretto, l'esecuzione del programma:

```
1 int main(void) {
2     struct chart *c = construct_chart();
3
4     set_song_title(c, "O luna tua", 3);
5     set_song_title(c, "Canzone stonata", 1);
6     set_song_title(c, "Red submarine", 8);
7
8     print_chart(c);
9
10    return 0;
11 }
```

dovrà stampare:

```
1 song list
2 [1] Canzone stonata
3 [2] -
4 [3] O luna tua
5 [4] -
6 [5] -
7 [6] -
8 [7] -
9 [8] Red submarine
10 [9] -
11 [10] -
```