

FONDAMENTI DI INFORMATICA I
Corso di Laurea in Ingegneria Elettronica, Informatica, delle
Telecomunicazioni
Università La Sapienza - Consorzio Nettuno
Appello del 13-03-2004
Parte Pratica (20 punti)

Esercizio 1 (5 punti)

Si scriva la funzione C che usa un ciclo **while** e che:

- riceve in input i due parametri **V**, puntatore ad una stringa, ed **N** intero
- inverte l'ordine dei caratteri di **V** **eseguendo al più N/2 cicli**

Esempio: la stringa di **13** caratteri "**la brutta zia**" diventa "**aiz atturb al**" con al più **6** cicli.

Esercizio 2 (5 punti)

Si scriva una funzione C **esercizio2** che conta il numero di parole lette da standard input.

- una parola è una sequenza di caratteri alfanumerici (lettere e numeri) consecutivi
- dovete usare la funzione **getchar**, che torna un intero pari al codice ASCII del prossimo carattere letto oppure **-1** se si è arrivati ad EOF

Esercizio 3 (5 punti)

Si scriva la funzione C che riceve come argomento il numero intero **M minore o uguale a 100** e:

- legge da standard input una successione di numeri interi separati da spazi o \n terminata dal numero **0**
- per ogni numero letto calcola la media degli **ultimi M valori positivi** letti e la stampa su una nuova riga con 2 decimali di precisione.

Nota: Quando il numero di interi positivi letti è minore di **M** non stampa niente.

Ad esempio, con **M=5** la sequenza: **3 6 -5 9 -9 1 -1 -5 -7 3 6 -7 2 -5 0**

Produrrebbe la sequenza di numeri (su righe successive): **4.40 5.00 5.00 4.20**

Si dia un paio di esempi di funzionamento (numeri in input e corrispondenti numeri in output)

Esercizio 4 (5 punti)

- Si svolga l'**esercizio 2** usando solo la ricorsione e lo stesso numero di chiamate.
- Si dettagli con chiarezza un esempio che mostri la sequenza di chiamate/uscite della funzione su almeno 2 esempi di input.

Parte teorica (10 punti)

Esercizio 5

Si supponga di dover progettare un circuito di controllo per un ascensore con memoria in un palazzo di 3 piani ($N=1,2,3$) che dispone dei seguenti sensori:

- A: porta ascensore aperta
- B(N): ascensore al piano N
- C: persone presenti all'interno
- D(N): bottone di destinazione N premuto (questo è l'output della memoria)
- E: bottone di allarme premuto
- F(N): bottone di chiamata N premuto (questo è l'output della memoria)
- G: l'ascensore si muove in su

Si scrivano le due formule booleane X ed Y che sono vere se solo se:

- $X(N)$ = aprire le porte al piano N è sicuro
- Y = l'ascensore deve muoversi (ma non se sarebbe uno spreco farlo)

Esercizio 6

Sia X un numero intero non negativo rappresentato da 3 bit $\{x_2, x_1, x_0\}$ nella codifica binaria naturale.

Si vuol costruire un circuito che calcola il numero $Y = (X^2/3) - 8$, codificato con una stringa di 4 bit $Y = \{y_3, y_2, y_1, y_0\}$ nella codifica in complemento a 2 (nella formula si usi la divisione intera).

Si scriva la tabella di verità del circuito, tenendo conto che i valori fuori dal range di rappresentazione vanno rappresentati col valore $Y = -1$

Soluzioni

Esercizio 1

```
void esercizio1(char * stringa, int N) {
    char c;
    int i = 0;
    while (i<N/2) {
        c = stringa[i];
        stringa[i] = stringa[N-i-1];
        stringa[N-i-1] = c;
        i++;
    }
}
```

Esercizio 2

```
#include <stdio.h>
#define FALSE 0
#define TRUE 1
#define EOF -1
int esercizio2() {
    int parolaP = FALSE; // inizio da non-parola
    int num_parole = 0;
    char c;
    while ((c=getchar())!=EOF) { // leggo il carattere
        if ('0'<=c && c<='9' ||
            'a'<=c && c<='z' ||
            'A'<=c && c<='Z') {
            if (!parolaP)
                num_parole++; // conto le trasizioni non-parola -> parola
            parolaP = TRUE; // ricordo che sono in una parola
        } else {
            parolaP = FALSE; // ora sono fuori da una parola
        }
    }
    return num_parole;
}
```

Esercizio 3

```
#include <stdio.h>
#define MAXM 100
int esercizio3(int M) {
    int numeri[MAXM] = { 0 }; // devo ricordare al massimo MAXM valori
    int quanti = 0; // quanti ne ho letti
    int numero, i, somma; // variabili varie
    scanf("%d",&numero); // leggo il primo numero
    while (numero != 0) {
        if (numero>0) { // gestisco una coda di M valori
            numeri[quanti % M] = numero;
            quanti++;
            if (quanti >= M) { // se ne ho letti abbastanza
                somma = 0;
                for (i=0;i<M;i++)
                    somma += numeri[i]; // li sommo
                printf("%.2f\n", (double)somma/M); // stampo la media
            }
        }
        scanf("%d",&numero);
    }
    return num_parole;
}
```

Esercizio 4

```
void esercizio4(char * stringa, int N) {
```

```

char c;
// se sono meno di 2 la stringa è a posto così
if (N<2)
    return;
// scambio il primo e l'ultimo carattere
c = stringa[0];
stringa[0] = stringa[N-1];
stringa[N-1] = c;
// chiamata ricorsiva per invertire la parte centrale
esercizio4(stringa+1,N-2);
}

```

Esercizio 5

Qualsiasi tentativo di svolgere *decentemente* l'esercizio veniva accettato, ad esempio:

```

X(N) = B(N) // basta che l'ascensore sia al piano N
Y = not A ( // porta chiusa
    G ( // direzione SU
        B(1) ( A (D(2) + D(3)) + F(2) + F(3) ) // da 1 va a un piano superiore
        +
        B(2) ( A D(3) + F(3) ) // da 2 va a un piano superiore
    )
    +
    not G(
        B(3) ( A (D(2) + D(1)) + F(2) + F(1) ) // da 3 va a un piano inferiore
        +
        B(2) ( A D(1) + F(1) ) // da 2 va a un piano inferiore
    )
)

```

Esercizio 6

I numeri da rappresentare sono: -8, -8, -7, -5, -3, 0, 4, 8

Dato che il range di rappresentazione di Y è da -8 a 7 il numero 8 va sostituito col numero -1.

X	x_2	x_1	x_0	Y	y_3	y_2	y_1	y_0
0	0	0	0	-8	1	0	0	0
1	0	0	1	-8	1	0	0	0
2	0	1	0	-7	1	0	0	1
3	0	1	1	-5	1	0	1	1
4	1	0	0	-3	1	1	0	1
5	1	0	1	0	0	0	0	0
6	1	1	0	4	0	1	0	0
7	1	1	1	8 (-1)	1	1	1	1