

# Laboratorio di Elementi di Architetture e Sistemi Operativi

Esercizi del 2 Maggio 2012

**Esercizio 1.** Scrivere un programma che dichiari le seguenti variabili ed esegua le operazioni specificate di seguito:

```
double tassointeresse = 7.3, tassosconto;  
double *dblptr;
```

1. Stampa il valore di `tassointeresse` sullo schermo;
2. Assegna l'indirizzo della variabile `tassointeresse` al puntatore `dblptr`;
3. Stampa il valore puntato da `dblptr` sullo schermo;
4. Assegna il valore puntato da `dblptr` alla variabile `tassosconto`;
5. Stampa il valore di `tassosconto` sullo schermo;
6. Stampa l'indirizzo di `tassointeresse` sullo schermo.
7. Stampa l'indirizzo di `tassosconto` sullo schermo.
8. Stampa l'indirizzo memorizzato in `dblptr` sullo schermo.
9. Il valore stampato è lo stesso dell'indirizzo di `tassointeresse`? Oppure è lo stesso dell'indirizzo di `tassosconto`?

Per stampare un indirizzo di memoria usare la direttiva `%p` di `printf`.

**Esercizio 2.** Scrivete una funzione con prototipo

```
void split-time ( long int sec, int *h, int *m, int *s )
```

che, dato un orario fornito in numero di secondi dalla mezzanotte (00:00:00), calcoli l'orario equivalente in ore, minuti, secondi, e lo memorizzi nelle tre variabili puntate da `h`, `m` e `s` rispettivamente. Scrivere un programma di esempio che usi la funzione.

**Esercizio 3.** Scrivete una funzione con prototipo

```
int *smallest( int a[], int n )
```

che, dato un vettore `a` di lunghezza `n`, restituisca un puntatore all'elemento più piccolo dell'array. Scrivere un programma di esempio che usi la funzione.

**Esercizio 4.** Una parola *palindroma* è una parola che, letta a rovescio, rimane identica. Scrivete una funzione che stabilisca se il suo argomento è una parola palindroma oppure no, usando due puntatori per scorrere la parola partendo dall'inizio e dalla fine, e *senza usare le funzioni di libreria per le stringhe*. Quindi scrivete un programma che stabilisca, per ciascun argomento *fornito da linea di comando*, se si tratta di una parola palindroma oppure no.

**Esercizio 5.** Scrivere le seguenti funzioni, utilizzando i puntatori e *senza usare le funzioni di libreria per le stringhe*:

- `void concat(char *s, char *t)` che aggiunge la stringa `t` al termine della stringa `s` (supponete che la lunghezza di `s` sia sufficiente a contenere la concatenazione di `s` e `t`);
- `void reverse(char *s)` che inverte la stringa `s` (e.g. `abc`  $\rightsquigarrow$  `cba`);

Utilizzate le funzioni `strcat` e `reverse` per scrivere i seguenti comandi:

1. un comando `uniscipar` che unisce in una riga sola tutti i paragrafi di un testo, eliminando le linee vuote. Un paragrafo è un blocco di righe di testo separato dagli altri da una o più linee vuote;
2. un comando `inverti` che scrive alla rovescia le righe di un testo.

Ogni comando deve leggere l'input da un file passato come *primo argomento* del `main`, e stampare il risultato sullo schermo. Si supponga che le linee del testo di output non siano mai più lunghe di 100 caratteri.

**Esempio:** File input.txt:

Questo è un testo  
diviso in paragrafi.

Ogni paragrafo è separato  
dagli altri da una o più  
linee vuote.

Fine del testo.

**Esecuzione dei comandi:**

```
$ uniscipar input.txt
Questo è un testo diviso in paragrafi.
Ogni paragrafo è separato dagli altri da una o più linee vuote.
Fine del testo.
```

```
$ inverti input.txt
otset nu è otseuQ
.ifargarap ni osivid
```

```
otarapes è ofargarap ingO
ùip o anu ad irtla ilgad
.etouv eenil
```

```
.otset led eniF
```