

# Laboratorio di Informatica Generale I UD

## Nona esercitazione

Danilo Severina

18 Maggio 2006

### Variabili globali

Le variabili **locali** (viste finora) hanno validità solo nelle funzioni in cui sono state definite.

Le variabili **globali** sono invece visibili da tutte le funzioni che appartengono al programma. Queste funzioni devono essere definite all'esterno di tutte le funzioni. La sintassi per la definizione è la stessa delle variabili locali.

### Allocazione dinamica

Funzione:

```
malloc(numero_byte_da_alloccare)
```

Esempi:

**variabile**

```
float* p=malloc(sizeof(float));  
unsigned char* s=malloc(sizeof(unsigned char));
```

**vettore**

```
int* v=malloc(sizeof(int)*n);  
char* s=malloc(sizeof(char)*n);
```

**matrice**

```
int** m;  
m=malloc(sizeof(int*)*n_righe);  
for(i=0 ; i<n_righe ; i++)  
    m[i]=malloc(sizeof(int)*n_colonne);
```

## Esercizi

### Esercizio 9.1

Scrivere un programma che chieda all'utente un numero  $n$  e inizializzi un vettore di interi di dimensione pari ad  $n$  con valori casuali. Utilizzare allocazione dinamica per il vettore.

### Esercizio 9.2

Scrivere un programma che dato un vettore di numeri casuali (esercizio precedente) sia in grado di riordinare il vettore. Il vettore può contenere più occorrenze dello stesso valore: nel vettore ordinato si devono mantenere tutti i valori.

Struttura main:

- inizializza vettore  $v$
- stampa vettore  $v$
- ordinare vettore  $v$
- stampa vettore  $v$

Struttura funzione ordinamento:

- ricerca del massimo nel vettore  $v$
- allocazione dinamica di vettore  $v2$  di dimensione pari al massimo valore trovato
- inizializzare elementi di  $v2$  a zero
- calcolo occorrenze elementi di  $v$  (memorizzare in  $v2$ )
- riscrittura del vettore  $v$  (ora ordinato)

### Esercizio 9.3

Scrivere un programma che richieda come parametri una stringa  $s$  ed un carattere  $c$  e sia in grado di restituire una stringa da cui sono state rimosse tutte le occorrenze del carattere inserito.

Struttura main:

- leggi stringa  $s$
- leggi carattere  $c$
- stampa  $s$
- manipolazione  $s$
- stampa  $s$

Per la manipolazione della stringa utilizzare un vettore temporaneo allocato dinamicamente di lunghezza opportuna.

## Esercizio 9.4

Scrivere un programma in grado di gestire un semplice vocabolario. Il programma deve essere in grado di:

- inserire nuove parole
- cancellare parole presenti
- stampare tutte le parole presenti
- terminare.

Per la memorizzazione delle parole utilizzare un vettore di stringhe definita come variabile globale. Il numero massimo di parole è definito a priori, mentre la lunghezza di ogni singola parola deve essere memorizzata in una stringa di dimensione opportuna (allocazione dinamica).

Struttura main:

- leggere operazione da eseguire:
  - se inserimento
    - leggi stringa s
    - insert(s)
  - se cancellazione
    - leggi stringa s
    - cancel(s)
  - se stampa
    - print()
  - se fine
    - print()
    - cancelAll()
    - fine