# ZZZ01 Esercizi Vari

Esercizi per preparazione alla prova pratica di laboratorio

# Esercizio ZZZ01_01 - **Elefanti Schizzinosi**

Nella savana africana c'e' una pozza d'acqua fresca alimentata da un acquedotto. La pozza può contenere al massimo 500 litri d'acqua.
C'e' un branco di N=5 elefanti che bevono ciascuno ogni volta 100 litri impiegando ciascuno 1/10 di secondo. Dopo avere bevuto ogni elefante fa un giretto e torna a bere dopo 1 secondo.
C'e' un branco di M=10 pekari che bevono ciascuno ogni volta 1 litro impiegando ciascuno 1/100 di secondo. Dopo avere bevuto ogni pekari fa un giretto e torna a bere dopo 2.2 secondi.
Potenzialmente alla pozza c'e' posto sufficiente affinché tutti gli animali possano bere assieme, però ci sono delle complicazioni.
Più pekari possono bere contemporaneamente ad altri pekari.
Più elefanti possono bere contemporaneamente ad altri elefanti.
Se uno o più elefanti sta bevendo, ed un pekari inizia a bere, gli elefanti completano la bevuta.
Se uno o più pekari stanno bevendo, nessun elefante si avvicina perché i pekari puzzano maledettamente, quindi gli elefanti aspettano.
Un animale può cominciare a bere solo se nella pozza c'è acqua sufficiente per
- far completare la bevuta di tutti gli animali che stanno già bevendo,
- ed anche a far completare la bevuta dell'animale che vuole cominciare a bere.

Ogni secondo, ed impiegando un tempo infinitesimo, l'acquedotto aggiunge alla pozza 70 litri. Se la pozza è già abbastanza piena, l'acqua in eccesso fuoriesce dalla pozza e viene persa
**Modellare ed implementare il sistema descritto**, utilizzando dei thread POSIX per ciascuna figura (acquedotto, pekari, elefanti) ed avvalendosi delle opportune strutture dati per la sincronizzazione. Scrivere immediatamente il Makefile per compilare e linkare i sorgenti. **Occorre inserire il controllo di errore nelle chiamate a funzione delle librerie dei pthread. In caso di errore grave, terminare il programma producendo un avviso a video.**

# Esercizio ZZZ01_02 - Macro ProdottoAumentato

In linguaggio ANSI C, scrivere una macro avente nome ProdottoAumentato che prende due argomenti. Gli argomenti passati alla macro possono essere di qualunque tipologia.
La macro deve incrementare di 1 il primo argomento e deve moltiplicare il risultato dell'incremento per il secondo argomento.
Scrivere un file main.c che contenga un piccolo programma di esempio in cui viene invocata la macro.
Scrivere anche un Makefile per compilare  e linkare il programma.

# Esercizio ZZZ01_03 - Makefile

In linguaggio ANSI C, scrivere una semplice programma che stampa a video la frase
°ciao, come stai?" e poi va a capo riga e termina.
Devono essere stampati anche i due doppi apici ad inizio e fine della frase.

Scrivere un semplice Makefile che deve far eseguire l'eseguibile ottenuto da compilazione e linking dei sorgenti del programma.

# Esercizio ZZZ01_04 - Stringhe

In linguaggio ANSI C, scrivere una semplice programma (ed il Makefile per compilarlo e linkarlo) che deve svolgere questi semplici compiti:

- Inizializzare una stringa con il contenuto "ghini non sa andare in bicicletta".
Stampare a video il contenuto della stringa.

- Poi Modificare il contenuto della stringa sostituendo con 3 caratteri bianchi (space, ' ', codice ascii 32, 0x20) ciascuno dei caratteri che compongono la sottostringa "non".
Stampare a video il nuovo contenuto della stringa.

- Poi Invertire il contenuto della stringa, ovvero mettere il primo carattere al posto dell'ultimo, il secondo carattere al posto del penultimo e così via.
Stampare a video il nuovo contenuto della stringa.

- Ridurre la lunghezza del contenuto della stringa, eliminando dal contenuto tutti i caratteri 'c'.
Stampare a video il nuovo contenuto della stringa.

# Esercizio ZZZ01_05 - Stringhe

In linguaggio ANSI C, sia dato il seguente programma :

```
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
int main(void)        {
            char  stringa[120] = "ciao";
            char  *p;

            p= (char*) malloc ( strlen(stringa) +1);
            if( !p )  {   printf("malloc failed\n");    return(1);  }
            strcpy( p, stringa );
            printf( "%s\n" , p );
            free( p );
            free ( stringa );
            return(0);
}
```

**Il programma viene compilato e linkato correttamente oppure produce errori?**

**Se credete che l'eseguibile  possa essere generato, eseguendo l'eseguibile accade qualche errore a run-time oppure no?**

# Esercizio ZZZ01_06 - bash

Scrivere uno script bash **lanciaedopokilla.sh** che lancia 10 volte direttamente in background il comando sleep(1000). Durante il lancio dei comandi NON deve essere memorizzato il pid dei processi lanciati in background.
Solo DOPO avere lanciato TUTTI i 10 comandi, utilizzare i comandi bash più opportuni per rintracciare i pid dei processi sleep in esecuzione ed eliminare tali processi.

# Esercizio ZZZ01_07 - bash

Scrivere uno script bash **cerca.sh** che cerca tutti i file con estensione .h nella directory /usr/include/linux/netfilter/ cercando anche nelle sue sottodirectory.
Per ciascun file trovato controllare se in quel file è presente la stringa int
Stampare a video il nome (inteso come percorso assoluto) di quei file che NON contengono la stringa int

Soluzione in pagina successiva

# soluzione Esercizio ZZZ01_07 - bash

```
for name in `find /usr/include/linux/netfilter -type f -name "*.h" -print` ; do NUMLINES=`grep int ${name} | wc -l` ; if (( ${NUMLINES} == 0 )) ; then echo ${name} ; fi ; done ;
```