# Prova scritta di Sistemi di Elaborazione
# Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

19 febbraio 2010

1. Supponiamo di avere il seguente programma scritto in parte in Assembler e in parte in C++:

```
        .data
        fmt: .asciz "%d"
        .text
        .global f3
f3:     pushl %ebp
        movl  %esp, %ebp
        subl  $4, %esp
        pushl %ebx
        pushl %ecx
        pushl %edx
        pushl %esi
        movl  8(%ebp), %ebx
        movl  $0, %esi
l1:     cmpl  12(%ebp), %esi
        jge   l2
        leal  -4(%ebp), %ecx
        pushl %ecx
        pushl $fmt
        call  scanf
        addl  $8, %esp
        cmpl  $0, -4(%ebp)
        jl    l2
        movl  -4(%ebp), %edx
        movl  %edx, (%ebx, %esi, 4)
        incl  %esi
        jmp   l1
l2:     movl  %esi, %eax
        popl  %esi
        popl  %edx
        popl  %ecx
        popl  %ebx
        leave
        ret
```

```
#include <stdio.h>

int f3(int* p, int n);

void f2(int v[], int x, int y) {
  int tmp = v[x];
  v[x] = v[y];
  v[y] = tmp;
}

void f1(int v[], int n){
  int i, j;
  for(i=0; i<n-1; i++) {
    int ind = i;
    for(j=i+1; j<n; j++)
      if(v[j]>v[ind])
        ind = j;
    f2(v, ind, i);
  }
}

int main(int argc, char* argv[]) {
  const int MAX = 100;
  int i, k;
  int aa[MAX];
  k = f3(aa, MAX);
  f1(aa, k);
  for(i=0; i<k; i++)
    printf("%d\n", aa[i]);
  return 0;
}
```

(a) Dire cosa viene calcolato dal programma complessivo.

(b) Tradurre la funzione `f1` in Assembler.

2. Scrivere i seguenti programmi in C++, utilizzando le primitive di Unix e la libreria standard del C.

   (a) Un programma `testa` con zero o più argomenti da riga di comando. Gli argomenti sono nomi di file. Il programma deve mostrare sull'uscita standard le prime dieci righe di ogni file, nell'ordine in cui i loro nomi sono stati passati da riga di comando (se un file non esiste o non è leggibile deve stampare un messaggio di errore e passare al prossimo). Se non vi sono nomi di file, il programma deve mostrare le prime 10 righe del suo ingresso standard. Il primo argomento, se è presente e rappresenta un numero intero, è speciale: indica il numero di linee da mostrare per ciascuno file (o ingresso standard), al posto del valore 10 di default. In questo caso il numero deve essere maggiore di 0.

   (b) Un programma `numera` con zero o più nomi di file come argomenti da riga di comando. Il programma deve mostrare le prime 20 righe di ciascun file (o dell'ingresso standard, se nessun nome è stato passato) numerate progressivamente a partire da 1. Per svolgere il suo compito il programma deve creare due processi collegati tramite pipe. Il primo processo deve eseguire il programma `testa` con argomenti opportuni, mentre il secondo processo deve eseguire il programma di sistema `nl` con argomento "`-na`". Il programma principale deve quidi attendere la terminazione dei due processi e terminare esso stesso.