

## Arhitectura sistemelor de calcul – Laboratorul 4

### Probleme cu proceduri

1. Sa se implementeze procedura `cifre_pare(x)`, cu  $x$  numar natural nenul, care afiseaza pe ecran cifrele pare ale numarului  $x$  (in orice ordine).
2. Sa se implementeze procedura `prim(x)`, cu  $x$  numar intreg, care decide daca numarul  $x$  este prim sau nu. (in  $\$v0$  ca `.word`)
3. Sa se implementeze procedura `perfect(x)`, cu  $x$  numar natural nenul, care decide daca numarul  $x$  este perfect sau nu. (in  $\$v0$ , ca `.word`)
4. Sa se implementeze procedura `divizibile_in_interval(a, b, x)` care determina cate numere sunt divizibile cu  $x$  in intervalul  $[a, b]$ , stiind ca  $a < b$  si ca  $a, b$ , si  $x$  sunt numere naturale nenule. (in  $\$v0$ , ca `.word`)
5. Sa se implementeze un program MIPS care sa simuleze apelul  $f(g(x))$  pentru un  $x$  numar intreg dat.  $f(x) = 2 * x$  iar  $g(x) = x + 1$ . Se va implementa functia  $g(x)$ , apoi se va implementa functia  $f$ , care va apela intern functia  $g$ . Cum poate fi implementat acest comportament pentru functiile imbricate?
6. Dat un vector in memorie, sa se implementeze procedura `modifica(*v, n)` care adauga fiecarui element valoarea 1. (de exemplu, vectorul  $[1, 2, 3]$  devine  $[2, 3, 4]$ ).
7. Dat un sir de caractere in memorie, sa se implementeze procedura `afiseaza_vocale(*str)`. Pentru aceasta, sa se implementeze o functie `este_vocala(ch)` care primeste un `.byte` si intoarce in  $\$v0$  valorile 1 sau 0, dupa cum `.byte-ul`  $ch$  este sau nu vocala.
8. Se citesc de la tastatura  $n$  (numar natural nenul) si un vector de  $n$  numere naturale (elemente intregi). Sa se scrie o procedura `numere_prime(*v, n)` care completeaza, intr-un spatiu din memorie destinat incarcarii unui vector, elementele prime din vectorul  $v$ . Pentru aceasta rezolvare, se va face un apel imbricat pentru a putea utiliza procedura `prim(x)`.
9. **(stil test de laborator)** Se citesc de la tastatura  $n$  (numar natural nenul) si un **vector** de  $n$  numere naturale (elemente intregi). Sa se scrie o *procedura* `evaluateaza_expresie(*v, n, x, y)` (cu  $x$  si  $y$  numere naturale nenule) care sa returneze in varful stivei rezultatul urmatoarei expresii (unde `suma_cif(z)` calculeaza suma cifrelor *numarului natural*  $z$ , iar **apelurile sunt imbricate** – adica in `evaluateaza_expresie` se face apel la `suma_cif`):

$$evaluateaza\_expresie(*v, n, x, y) := \sum_{i=0}^{n-1} \left[ (suma\_cif(v[i]) * x) - \left\lfloor \frac{i}{2} \right\rfloor * y \right]^2$$