

Laborator 10

Clips (2)

1. Scrieti un program care calculeaza suma elementelor unei liste.

```
(deffacts fapte (lista 1 2 3 4 5) (rez 0))
```

```
(defrule suma
```

```
  ?f1 <- (lista ?prim $?rest)
```

```
  ?f2 <- (rez ?r)
```

```
=>
```

```
  (retract ?f1)
```

```
  (retract ?f2)
```

```
  (assert (lista $?rest))
```

```
  (assert (rez (+ ?prim ?r)))
```

```
)
```

2. Scrieti un program care realizeaza reuniunea a 2 multimi. O multime este reprezentata ca o lista.

```
(deffacts fapte (lista1 1 2 3 4 5) (lista2 3 4 5 6 7) (rez))
```

```
(defrule reuniune
```

```
  (or (lista1 $? ?x $?) (lista2 $? ?x $?))
```

```
  (not (rez $? ?x $?))
```

```
  ?f <- (rez $?a)
```

```
=>
```

```
  (retract ?f)
```

```
  (assert (rez $?a ?x))
```

)

3. Scrieti un program care sorteaza crescator o lista prin interschimbari.

```
(deffacts fapte (lista 1 3 4 2 5))
```

```
(defrule sortare
```

```
  ?f <- (lista $?a ?x ?y&:(> ?x ?y) $?b)
```

```
=>
```

```
  (retract ?f)
```

```
  (assert (lista $?a ?y ?x $?b))
```

```
)
```

4. Scrieti un program care pentru o lista de numere naturale creeaza doua liste, una cu numerele de pe pozitiile pare si alta cu numerele de pe pozitiile impare.

```
(deffacts fapte (lista 1 2 3 4 5) (contor 1) (pare ) (impare ))
```

```
(defrule imparte
```

```
  ?f1 <- (lista ?prim $?rest)
```

```
  ?f2 <- (pare $?p)
```

```
  ?f3 <- (impare $?i)
```

```
  ?f4 <- (contor ?cnt)
```

```
=>
```

```
  (retract ?f1)
```

```
  (assert (lista $?rest))
```

```
  (retract ?f4)
```

```
  (assert (contor (+ ?cnt 1)))
```

```
  (if (eq (mod ?cnt 2) 0)
```

```
      then
```

```
        (retract ?f2)
```

```
                (assert (pare $?p ?prim))
            else
                (retract ?f3)
                (assert (impare $?i ?prim))
        )
    )
```

5. Scrieti un program care inlocuieste toate aparitiile unui numar x cu un numar y intr-o lista.

```
(deffacts fapte (lista 1 2 3 2 4 2) (e1 2) (e2 7))
```

```
(defrule inlocuire
```

```
    (e1 ?x)
```

```
    (e2 ?y)
```

```
    ?f <- (lista $?a ?x $?b)
```

```
=>
```

```
    (retract ?f)
```

```
    (assert (lista $?a ?y $?b))
```

```
)
```