

# Programul MIPS modificat

- B"000001\_00000\_00001\_0000000000000001", --0 (X4010001) addi \$1 , \$0 , 1 => initializare contor parcurgere in registrul 1 cu valoarea 1
- B"000001\_00000\_00010\_0000000000000000", --1 (X4020000) addi \$2 , \$0 , 0 => index locatie memorie elemente din vector in registrul 2 cu valoarea 0
- B"000010\_00000\_00011\_00000000000000100", --2 (X8030004) lw \$3 , 4(\$0) => se incarca N in registrul 3 din memorie de la adresa 4
- B"000000\_11111\_11111\_11111\_00000\_000000", --3 NoOp
- B"000000\_11111\_11111\_11111\_00000\_000000", --4 NoOp
- B"000000\_00000\_00011\_00100\_00000\_000000", --5 (X32000) add \$4 , \$0 , \$3 => punem in registrul 4 continutul registrului 3 ( numarul maxim de iteratii)
- B"000001\_00000\_00101\_0000000000000000", --6 (X4050000) addi \$5 , \$0 , 0 => in registrul 5 initializam maximul numerelor pare cu 0
- B"000001\_00000\_01000\_0000000000000000", --7 (X4080000) addi \$8 , \$0 , 0 => un flag pentru slt , initializat cu 0
- B"000100\_00001\_00100\_00000000000011001", --8 (10240019) beq \$1 , \$4 , 25 => verificam daca contorul este egal cu numarul maxim de iteratii ( in registrul 1 este contorul si in 4 numarul maxim de iteratii ) , iar in cazul in care sunt egale se sare la instructiunea 17
- B"000000\_11111\_11111\_11111\_00000\_000000", --9 NoOp
- B"000000\_11111\_11111\_11111\_00000\_000000", --10 NoOp
- B"000000\_11111\_11111\_11111\_00000\_000000", --11 NoOp
- B"000010\_00010\_00110\_0000000000001100", --12 (X846000C) lw \$6 , 12(\$2) => incarcam in registrul 6 elementul din vector de la adresa 12 + offsetul dat de registrul 2 ( aici vom incarca la fiecare iteratie urmatorul element din vector)
- B"000001\_00000\_01010\_00000000000000001", --13 (X40A0001) addi \$10 , \$0 , 1 => punem in registrul 10 valoarea 1
- B"000000\_11111\_11111\_11111\_00000\_000000", --14 NoOp
- B"000000\_11111\_11111\_11111\_00000\_000000", --15 NoOp
- B"000000\_00110\_01010\_00111\_00000\_000100", --16 (XCA3804) and \$7 , \$6 , \$10 => facem and intre registrul 6 adica numarul actual din vector si registrul 10 adica 1 pentru a afla paritatea
- B"000000\_11111\_11111\_11111\_00000\_000000", --17 NoOp

- B"000000\_11111\_11111\_11111\_00000\_000000", --18 NoOp
- B"000100\_00111\_01010\_0000000000001011", --19 (10EA000B) beq \$7 , \$10 , 11 => daca numarul nu este par se sare la instrutiunea 14
- B"000000\_11111\_11111\_11111\_00000\_000000", --20 NoOp
- B"000000\_11111\_11111\_11111\_00000\_000000", --21 NoOp
- B"000000\_11111\_11111\_11111\_00000\_000000", --22 NoOp
- B"000000\_00101\_00110\_01000\_00000\_000110", --23 (XA64006) slt \$8 , \$5 , \$6 => daca maximul este mai mic decat valoarea curenta atunci in registrul 8 se va afla valoarea 1 , in caz contrar va fi 0 (in registrul 5 avem maximul , iar in registrul 6 avem numarul din vector)
- B"000000\_11111\_11111\_11111\_00000\_000000", --24 NoOp
- B"000000\_11111\_11111\_11111\_00000\_000000", --25 NoOp
- B"000100\_01000\_00000\_0000000000000100", --26 (X11000004) beq \$8 , \$0 , 4 => daca valoarea din 8 este egala cu 0 atunci maximul nu se va schimba si se sare la instrutiunea 14
- B"000000\_11111\_11111\_11111\_00000\_000000", --27 NoOp
- B"000000\_11111\_11111\_11111\_00000\_000000", --28 NoOp
- B"000000\_11111\_11111\_11111\_00000\_000000", --29 NoOp
- B"000000\_00110\_00000\_00101\_00000\_000000", --30 (XC02800) add \$5 , \$6 , \$0 => in caz in care in registrul 8 avem valoarea unu , vom pune in registrul 5 ( unde este continut maximul) , suma dintre registrul 6 ( numarul curent din vector) si registrul 0 ( cu valoare constanta 0)
- B"000001\_00010\_00010\_0000000000000100", --31 (X4420004) addi \$2 , \$2 , 4 => adaugam la registrul 2 numarul 4 pentru a putea trece la urmatorul element din vector
- B"000001\_00001\_00001\_0000000000000001", --32 (X4210001) addi \$1 , \$1 , 1 => crestem contorul afla la registrul 1 cu 1
- B"000111\_000000000000000000000001000", --33 (X1C000008) j 8 => facem jump la instrutiunea 8
- B"000000\_11111\_11111\_11111\_00000\_000000", --34 NoOp
- B"000011\_00000\_00101\_0000000000001000", --35 (XC050006) sw \$5 , 8(\$0) => dupa ce se termina toata parcurgerea salvam continutul registrului 5 ( maximul rezultat) la adresa 8
- B"000010\_00000\_01100\_0000000000001000", --36 (X80C0008) lw \$12 , 8(\$0) => incarcam in registrul 12 ce am salvat la adresa 8 pentru a putea fi vizualizat in mem\_data