

Parcurgere problema

--0 addi \$1 , \$0 , 1 (Valoare in hexa pentru a urmari pe SSD : x"4010000")

B"000001_00000_00001_0000000000000001"

=> initializare contor parcurgere in registrul 1 cu valoarea 1

--1 addi \$2 , \$0 , 0 (Valoare in hexa pentru a urmari pe SSD : x"4020000")

B"000001_00000_00010_0000000000000000"

=> index locatie memorie elemente din vector in registrul 2 cu valoarea 0

--2 lw \$3 , 4(\$0) (Valoare in hexa pentru a urmari pe SSD : x"8030004")

B"000010_00000_00011_00000000000000100"

=> se incarca N in registrul 3 din memorie de la adresa 4

--3 add \$4 , \$0 , \$3 (Valoare in hexa pentru a urmari pe SSD : x"32000")

B"000000_00000_00011_00100_00000_000000"

=> punem in registrul 4 continutul registrului 3 (numarul maxim de iteratii)

--4 addi \$5 , \$0 , 0 (Valoare in hexa pentru a urmari pe SSD : x"4050000")

B"000001_00000_00101_0000000000000000"

=> in registrul 5 initializam maximul numerelor pare cu 0

--5 addi \$8 , \$0 , 0 (Valoare in hexa pentru a urmari pe SSD : x"4080000")

B"000001_00000_01000_0000000000000000"

=> un flag pentru slt , initializat cu 0

--6 beq \$1 , \$4 , 10 (Valoare in hexa pentru a urmari pe SSD : x"1024000A")

B"000100_00001_00100_0000000000001010"

=> 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

--7 lw \$6 , 12(\$2) (Valoare in hexa pentru a urmari pe SSD : x"846000C")

B"000010_00010_00110_0000000000001100"

=> 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)

--8 addi \$10 , \$0 , 1 (Valoare in hexa pentru a urmari pe SSD : x"40A0001")

B"000001_00000_01010_0000000000000001"

=> punem in registrul 10 valoarea 1

--9 and \$7 , \$6 , \$10 (Valoare in hexa pentru a urmari pe SSD : x"CA3804")

B"000000_00110_01010_00111_00000_000100"

=> facem and intre registrul 6 adica numarul actual din vector si registrul 10 adica 1 pentru a afla paritatea

--10 beq \$7 , \$10 , 3 (Valoare in hexa pentru a urmari pe SSD : x"10EA0003")

B"000100_00111_01010_0000000000000011"

=> daca numarul nu este par se sare la instructiunea 14

--11 slt \$8 , \$5 , \$6 (Valoare in hexa pentru a urmari pe SSD : x"A64006")

B"000000_00101_00110_01000_00000_000110"

=> 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)

--12 beq \$8 , \$0 , 1 (Valoare in hexa pentru a urmari pe SSD : x"11000001")

B"000100_01000_00000_00000000000000001"

=> daca valoarea din 8 este egala cu 0 atunci maximul nu se va schimba si se sare la instructiunea 14

--13 add \$5 , \$6 , \$0 (Valoare in hexa pentru a urmari pe SSD : x"C02800")

B"000000_00110_00000_00101_00000_000000"

=> 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)

--14 addi \$2 , \$2 , 4 (Valoare in hexa pentru a urmari pe SSD : x"4420004")

B"000001_00010_00010_00000000000000100"

=> adaugam la registrul 2 numarul 4 pentru a putea trece la urmatorul element din vector

--15 addi \$1 , \$1 , 1 (Valoare in hexa pentru a urmari pe SSD : x"4210001")

B"000001_00001_00001_00000000000000001"

=> crestem contorul afla la registrul 1 cu 1

--16 j 6 (Valoare in hexa pentru a urmari pe SSD : x"1C000006")

B"000111_000000000000000000000000110"

=> facem jump la instructiunea 6

--17 sw \$5 , 8(\$0) (Valoare in hexa pentru a urmari pe SSD : x"C050006")

B"000011_00000_00101_000000000000001000"

=> dupa ce se termina toata parcurgerea salvam continutul registrului 5 (maximul rezultat) la adresa 8

--18 lw \$12, 8(\$0) (x"80C0008")

B"000010_00000_01100_0000000000001000"

=> incarcam in registrul 12 ce am salvat la adresa 8 pentru a putea fi vizualizat in mem_data