

Tabella1

Codifica dei primi 34 programma MRI

0	1	2	3	4	5	6
Z(1)	S(1)	Z(1) Z(1)	T(1,1)	Z(1) S(1)	S(1) Z(1)	Z(1) Z(1) Z(1)
7	8	9	10	11	12	13
C(1,1,1)	Z(1) T(1,1)	S(1) S(1)	Z(1) Z(1) S(1)	T(1,1) Z(1)	Z(1) S(1) Z(1)	S(1) Z(1) Z(1)
14	15	16	17	18	19	20
Z(1) Z(1) Z(1) Z(1)	Z(2)	Z(1) C(1,1,1)	S(1) T(1,1)	Z(1) Z(1) T(1,1)	T(1,1) S(1)	Z(1) S(1) S(1)
21	22	23	24	25	26	27
S(1) Z(1) S(1)	Z(1) Z(1) Z(1) S(1)	C(1,1,1) Z(1)	Z(1) T(1,1) Z(1)	S(1) S(1) Z(1)	Z(1) Z(1) S(1) Z(1)	T(1,1) Z (1) Z (1)
28	29	30	31	32	33	34
Z (1) S (1) Z (1) Z (1)	S (1) Z (1) Z (1) Z (1)	Z (1) Z (1) Z (1) Z (1) Z (1)	S(2)	Z(1) Z(2)	S(1) C(1,1,1)	Z(1) Z(1) C(1,1,1)

java urm program [param1 [param2 [...]]] [time steplimit]

program is either an integer representing the program's code number (as described in chapter 4 of Cutland's text) or the name of a text file containing the program's listing.

param1, param2, ... are input values supplied to the URM program, i.e., the initial values of the first few registers in the register bank. By default, all registers begin with an initial value of zero.

steplimit, if specified, is the maximum number of steps the URM processor should execute before "giving up" on the program. This parameter is useful for situations in which the input vector may not be in the domain of the program (i.e., the program does not halt when started with the given input).

Esempio 1

```
>java urm 9007203549970431
Decoding program...done.
Executing program...done.
```

Execution summary:

```
-----
program: 9007203549970431
input: none
output: 0
time: 3 steps
```

Program listing:

```
-----
1: T (1, 3)  [#18]
2: S (4)    [#13]
3: Z (6)    [#20]
```

Program code:

```
-----
9007203549970431
```

Esempio 2

sia add.urm un file esterno

```
top: J (3, 2, end)
      S (1)
      S (3)
      J (1, 1, top)
end:
```

```
>java urm add.urm 51 8
Resolving labels...done.
Loading program...done.
Executing program...done.
```

Execution summary:

```
-----
program: add.urm
input: (51, 8)
output: 59
time: 33 steps
```

Program listing:

```
-----
1: J (3, 2, 5) [#73727]
2: S (1) [#1]
3: S (3) [#9]
4: J (1, 1, 1) [#3]
```

Program code:

```
-----
76090488...69396735
```

Esempio 3

The function of program #176538093190184139370036875193483265

We will execute this program on the input (10, 5), for no more than 100 steps.

```
>java urm 176538093190184139370036875193483265 10 5 time 100
Decoding program...done.
Executing program...aborted.
```

Execution summary:

```
-----
program: 176538093190184139370036875193483265
input: (10, 5)
output: unknown
time: not less than 100 steps
```

Program listing:

```
-----
1: S (1)  [#1]
2: J (1, 2, 4)  [#111]
3: J (1, 1, 1)  [#3]
```

Program code:

```
-----
176538093190184139370036875193483265
```