**Regular expression**

∑=(0,1)

Q. Start with 1 and end with 0

RE: 1(0+1)\*0

L={10,100,110,1……0}

Q: String having at most two 0’s

RE: 1\*(0+) 1\* (0+)1\*

L={00, 0,111,1,10,01,…..}

Q: String containing exactly 3 1’s

RE: 0\* 1 0\* 1 0\* 1 0\*

TU.Q: Regular expression for defining valid identifier like programming in C. [TU]

RE:

∑=(0,1…..9,a,b…z, A, B……..Z,\_)

RE: **(a|b|…..|z|A|B|…..|Z|\_).(a|b|….. z|A |B|…..|Z |\_|0|1|…|9)\***

**Two dimensional array**

∑=(0,1…..9,a,b…z, A, B……..Z, \_ ,[ ,])

1D\_Array:

**(a|b|…..|z|A|B|…..|Z|\_).(a|b|….. z|A |B|…..|Z |\_|0|1|…|9)\***

**.[.(1|2|3…|9). (0|1|2|3…|9)\*.]. [.(1|2|3…|9). (0|1|2|3…|9)\*.]**

TU.Q: Regular expression for defining valid one dimensional array like programming in C.

Valid: a[20], sum[100]

Invalid: aa[-30], ram[0], anup[rajan]

RE:

∑=(0,1…..9,a,b…z, A, B……..Z,\_,[,])

RE: **(a|b|…..|z|A|B|…..|Z|\_) (a|b|….. z|A |B|…..|Z|\_|0|1|…….|9)\*** . [(1|2|….|9) (0|1|2|….|9)\* ]

TU. Q: Regular expression for defining valid two dimensional array like programming in C.

∑=(0,1…..9,a,b…z, A, B……..Z,\_,[,])

RE: **(a|b|…..|z|A|B|…..|Z|\_) (a|b|….. z|A |B|…..|Z|\_|0|1|…….|9)\*** . [(1|2|….|9) (0|1|2|….|9)\* ] [(1|2|….|9) (0|1|2|….|9)\* ]

**Regular Definition**

Q: Regular definition for defining valid identifier like programming in C.

RE:

∑=(0,1…..9,a,b…z, A, B……..Z,\_)

RE: (a|b|…..|z|A|B|…..|Z|\_) (a|b|….. z|A |B|…..|Z|\_|0|1|…….|9)\*

RD:

Lc🡪 a|b|…..|z

Uc🡪 A|B|…..|Z

Digit🡪0|1|…….|9

Us🡪\_

RD: (Lc|Uc|Us) (Lc|Uc|Us|Digit)\*

TU.Q: Regular definition for defining valid one dimensional array like programming in C.

a[20], sum[100]

aa[-30], ram[0], anup[rajan]

RE:

∑=(0,1…..9,a,b…z, A, B……..Z,\_,[,])

RE: **(a|b|…..|z|A|B|…..|Z|\_) (a|b|….. z|A |B|…..|Z|\_|0|1|…….|9)\*** . [(1|2|….|9) (0|1|2|….|9)\* ]

RD:

Lc🡪 a|b|…..|z

Uc🡪 A|B|…..|Z

Digit🡪1|…….|9

Zero🡪0

Us🡪\_

Lb🡪[

Rb🡪]

RD: (Lc|Uc|Us) (Lc|Uc|Us|Digit|zero)\* Lb. Digit.(Digit|zero)\* Rb