**REGULAR EXPRESSION**

*Reserved Words*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C | Proposed Language | Regular Expression | Token | |
| main | Lead | (L)(e)(a)(d) | | Lead |
| int | Int | (I)(n)(t) | | Int |
| double | Double | (D)(o)(u)(b)(l)(e) | | Double |
| char | Char | (C)(h)(a)(r) | | Char |
| string | String | (S)(t)(r)(i)(n)(g) | | String |
| bool | Boolean | (B)(o)(o)(l)(e)(a)(n) | | Boolean |
| void | Null | (N)(u)(l)(l) | | Null |
| const | Let | (L)(e)(t) | | Let |
| - | Array | (A)(r)(r)(a)(y) | | Array |
| - | of | (o)f) | | Of |
| - | Task | (T)(a)(s)(k) | | Task |
| struct | Object | (O)(b)(j)(e)(c)(t) | | Object |
| printf | Say | (S)(a)(y) | | Say |
| scanf | Read | (R)(e)(a)(d) | | Read |
| if | If | (I)(f) | | If |
| Else if | Or | (O)(r) | | Or |
| else | Otherwise | (O)(t)(h)(e)(r)(w)(i)(s)(e) | | Otherwise |
| switch | Option | (O)(p)(t)(i)(o)(n) | | Option |
| case | State | (S)(t)(a)(t)(e) | | State |
| default | Default | (D)(e)(f)(a)(u)(l)(t) | | Default |
| break | Stop | (S)(t)(o)(p) | | Stop |
| continue | Skip | (S)(k)(i)(p) | | Skip |
| For | For | (F)(o)(r) | | For |
| While | Until | (U)(n)(t)(i)(l) | | Until |
|  | Loop | (L)(o)(o)(p) | | Loop |
| While | LoopIf | (L)(o)(o)(p)(I)(f) | | LoopIf |
| Do | Do | (D)(o) | | Do |
| Clrscr | Clear | (C)(l)(e)(a)(r) | | Clear |
| Return | Response | (R)(e)(s)(p)(o)(n)(s)(e) | | Response |
| - | Yes | (Y)(e)(s) | | Yes |
| - | No | (N)(o) | | No |
| - | is | (i)(s) | | Is |
| - | Start | (S)(t)(a)(r)(t) | | Start |
| - | End | (E)(n)(d) | | End |
| - | Var | (V)(a)(r) | | Var |
| - | Let | (L)(e)(t) | | Let |
| - | by | (b)(y) | | By |
|  |  |  | |  |

*Identifiers*

|  |  |
| --- | --- |
| Regular Expression | Token |
| (CapLet) | Id |

*Literals*

|  |  |
| --- | --- |
| Regular Expression | Token |
| (Digit) | Intlit |
| (Digit) (.) (Number) | Doublelit |
| (‘) (Caplet/Lowlet/Numbers) (‘) | Charlit |
| (“) (”) | Stringlit |
| (Yes/No) | Boollit |

*Reserved Symbols*

|  |  |  |
| --- | --- | --- |
| Reserved Symbols | Regular Expression | Token |
| + | (+) | + |
| - | (-) | - |
| \* | (\*) | \* |
| / | (/) | / |
| % | (%) | % |
| ++ | (+)(+) | ++ |
| -- | (-)(-) | -- |
| == | (=)(=) | == |
| != | (=)(!) | =! |
| > | (>) | > |
| < | (<) | < |
| >= | (>)(=) | => |
| <= | (<)(=) | =< |
| && | (&)(&) | && |
| || | (|)(|) | || |
| ! | (!) | ! |
| . | (.) | . |
| , | (,) | , |
| ; | (;) | ; |
| ‘ ’ | (‘)(’) | ‘ ’ |
| “” | (“)(”) | “” |
| : | (:) | : |
| ( ) | (()()) | ( ) |
| [ ] | ([)(]) | [ ] |
| @ | (@) | @ |