***LEXICAL TEST SCRIPT***

***Reserved Words***

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **Actual Output** | **Remarks** |
| Array | Lexical Analyzer Encountered 1 error/s.  Please try again.  Proper delimiter expected: Array on line 1 | Lexical Analyzer Encountered 1 error/s.  Please try again.  Proper delimiter expected: Array on line 1 | **OK** |
| Array˽ | Token = Array | Token = Array | **OK** |
| Array↵ | Token = Array | Token = Array | **OK** |
| Array: | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper -delimiter expected: Array on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: : on line 1 | **OK** |
| Array. | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: . on line 1 | **OK** |
| Array( | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: ( on line 1 | **OK** |
| Array? | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: ? on line 1 | **OK** |
| Array- | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: - on line 1 | **OK** |
| Array+ | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: + on line 1 | **OK** |
| Array, | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: , on line 1 | **OK** |
| Array| | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: | on line 1 | **OK** |
| Array[ | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: [ on line 1 | **OK** |
| Array; | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Array on line 1  Invalid input: ; on line 1 | **OK** |
| Arrayy | Lexical Analyzer Encountered 1 error/s.  Please try again.  Invalid Input: Arrayy on line 1 | Lexical Analyzer Encountered 1 error/s.  Please try again.  Invalid Input: Arrayy on line 1 | **OK** |
| Array1 | Lexical Analyzer Encountered 1 error/s.  Please try again.  Invalid Input: Array1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Please try again.  Invalid Input: Array1 on line 1 | **OK** |
| Boolean | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Boolean on line 1 | Lexical Analyzer Encountered 2 error/s.  Please try again.  Proper delimiter expected: Boolean on line 1 | **OK** |
| Boolean˽ | Token = Boolean | Token = Boolean | **OK** |
| Boolean↵ | Token = Boolean | Token = Boolean | **OK** |
| Boolean: | Lexical Analyzer Encountered 1 error/s.  Please try again.  Proper delimiter expected: Boolean on line 1 | Lexical Analyzer Encountered 1 error/s.  Please try again.  Proper delimiter expected: Boolean on line 1 | **OK** |
| Boolean. | Lexical Analyzer Encountered 1 error/s.  Token = Boolean  Invalid Input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Boolean  Invalid Input: . on line 1 | **OK** |
| Boolean< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: < on line 1 | **OK** |
| Boolean( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: ( on line 1 | **OK** |
| Boolean? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: ? on line 1 | **OK** |
| Boolean- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: - on line 1 | **OK** |
| Boolean+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: + on line 1 | **OK** |
| Boolean, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: , on line 1 | **OK** |
| Boolean| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: | on line 1 | **OK** |
| Boolean[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: [ on line 1 | **OK** |
| Boolean; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: ; on line 1 | **OK** |
| Boolean/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Boolean on line 1  Invalid input: / on line 1 | **OK** |
| Booleann | Lexical Analyzer Encountered 2 error/s.  Invalid input: Booleann on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: Booleann on line 1 | **OK** |
| Boolean1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: Boolean1 on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: Boolean1 on line 1 | **OK** |
| by | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: by on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: by on line 1 | **OK** |
| by˽ | Token = by | Token = by | **OK** |
| by↵ | Token = by | Token = by | **OK** |
| by: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: : on line 1 | **OK** |
| by. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: . on line 1 | **OK** |
| by( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: ( on line 1 | **OK** |
| by? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: ? on line 1 | **OK** |
| by- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: - on line 1 | **OK** |
| by+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: + on line 1 | **OK** |
| by, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: , on line 1 | **OK** |
| by| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: | on line 1 | **OK** |
| by[ | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: by on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: by on line 1  Invalid input: [ on line 1 | **OK** |
| by; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: by on line 1  Invalid input: ; on line 1 | **OK** |
| byy | Lexical Analyzer Encountered 1 error/s.  Invalid input: byy on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: byy on line 1 | **OK** |
| by1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: by1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: by1 on line 1 | **OK** |
| Char | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Char on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Char on line 1 | **OK** |
| Char˽ | Token = Char | Token = Char | **OK** |
| Char↵ | Token = Char | Token = Char | **OK** |
| Char: | Lexical Analyzer Encountered 1 error/s.  Invalid input: : on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: : on line 1 | **OK** |
| Char. | Lexical Analyzer Encountered 1 error/s.  Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: . on line 1 | **OK** |
| Char< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: < on line 1 | **OK** |
| Char( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: ( on line 1 | **OK** |
| Char? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: ? on line 1 | **OK** |
| Char- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: - on line 1 | **OK** |
| Char+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: + on line 1 | **OK** |
| Char, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: , on line 1 | **OK** |
| Char| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: | on line 1 | **OK** |
| Char[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: [ on line 1 | **OK** |
| Char; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: ; on line 1 | **OK** |
| Char/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Char on line 1  Invalid input: / on line 1 | **OK** |
| Charr | Lexical Analyzer Encountered 1 error/s.  Invalid input: Charr on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Charr on line 1 | **OK** |
| Char1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Char1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Char1 on line 1 | **OK** |
| Clear | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Clear on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Clear on line 1 | **OK** |
| Clear˽ | Token = Clear | Token = Clear | **OK** |
| Clear↵ | Token = Clear | Token = Clear | **OK** |
| Clear: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: : on line 1 | **OK** |
| Clear. | Lexical Analyzer Encountered 2 error/s.  Invalid input: .on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: .on line 1 | **OK** |
| Clear< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: < on line 1 | **OK** |
| Clear( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: ( on line 1 | **OK** |
| Clear? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: ? on line 1 | **OK** |
| Clear- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: - on line 1 | **OK** |
| Clear+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: + on line 1 | **OK** |
| Clear, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: , on line 1 | **OK** |
| Clear| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: | on line 1 | **OK** |
| Clear[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: [ on line 1 | **OK** |
| Clear; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: ; on line 1 | **OK** |
| Clear/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Clear on line 1  Invalid input: / on line 1 | **OK** |
| Clearr | Lexical Analyzer Encountered 1 error/s.  Invalid input: Clearr on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Clearr on line 1 | **OK** |
| Clear1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Clear1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Clear1 on line 1 | **OK** |
| Default | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Default on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Default on line 1 | **OK** |
| Default˽ | Token = Default | Token = Default | **OK** |
| Default↵ | Token = Default | Token = Default | **OK** |
| Default: | Lexical Analyzer Encountered 1 error/s.  Token = Default Invalid input: : on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Default Invalid input: : on line 1 | **OK** |
| Default. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: . on line 1 | **OK** |
| Default( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: ( on line 1 | **OK** |
| Default? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: ? on line 1 | **OK** |
| Default- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: - on line 1 | **OK** |
| Default+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: + on line 1 | **OK** |
| Default, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: , on line 1 | **OK** |
| Default| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: | on line 1 | **OK** |
| Default[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: [ on line 1 | **OK** |
| Default; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: ; on line 1 | **OK** |
| Default/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Default on line 1  Invalid input: / on line 1 | **OK** |
| Defaultt | Lexical Analyzer Encountered 1 error/s.  Invalid input: Defaultt on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Defaultt on line 1 | **OK** |
| Default1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Default1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Default1 on line 1 | **OK** |
| Do | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Do on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Do on line 1 | **OK** |
| Do˽ | Token = Do | Token = Do | **OK** |
| Do↵ | Token = Do | Token = Do | **OK** |
| Do: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: : on line 1 | **OK** |
| Do. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: . on line 1 | **OK** |
| Do( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: ( on line 1 | **OK** |
| Do? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: ? on line 1 | **OK** |
| Do- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: - on line 1 | **OK** |
| Do+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: + on line 1 | **OK** |
| Do, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: , on line 1 | **OK** |
| Do| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: | on line 1 | **OK** |
| Do[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: [ on line 1 | **OK** |
| Do; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Do on line 1  Invalid input: ; on line 1 | **OK** |
| Doo | Lexical Analyzer Encountered 1 error/s.  Invalid input: Doo on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Doo on line 1 | **OK** |
| Do1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Do1on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Do1on line 1 | **OK** |
| Double | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Double on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Double on line 1 | **OK** |
| Double˽ | Token = Double | Token = Double | **OK** |
| Double↵ | Token = Double | Token = Double | **OK** |
| Double: | Lexical Analyzer Encountered 1 error/s.  Token = Double Invalid input: : on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Double Invalid input: : on line 1 | **OK** |
| Double. | Lexical Analyzer Encountered 1 error/s.  Token = Double Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Double Invalid input: . on line 1 | **OK** |
| Double< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: < on line 1 | **OK** |
| Double( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: ( on line 1 | **OK** |
| Double? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: ? on line 1 | **OK** |
| Double- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: - on line 1 | **OK** |
| Double+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: + on line 1 | **OK** |
| Double, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: , on line 1 | **OK** |
| Double| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: | on line 1 | **OK** |
| Double[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: [ on line 1 | **OK** |
| Double; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: ; on line 1 | **OK** |
| Double/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Double on line 1  Invalid input: / on line 1 | **OK** |
| Doublee | Lexical Analyzer Encountered 1 error/s.  Invalid input: Doublee on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Doublee on line 1 | **OK** |
| Double1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Double1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Double1 on line 1 | **OK** |
| End | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: End on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: End on line 1 | **OK** |
| End˽ | Token = End | Token = End | **OK** |
| End↵ | Token = End | Token = End | **OK** |
| End: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: : on line 1 | **OK** |
| End. | Lexical Analyzer Encountered 1 error/s.  Token = End Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = End  Invalid input: . on line 1 | **OK** |
| End< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: < on line 1 | **OK** |
| End( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: ( on line 1 | **OK** |
| End? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: ? on line 1 | **OK** |
| End- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: - on line 1 | **OK** |
| End+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: + on line 1 | **OK** |
| End, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: , on line 1 | **OK** |
| End| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: | on line 1 | **OK** |
| End[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: [ on line 1 | **OK** |
| End; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: ; on line 1 | **OK** |
| End/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: End on line 1  Invalid input: / on line 1 | **OK** |
| Endd | Lexical Analyzer Encountered 1 error/s.  Invalid input: Endd on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Endd on line 1 | **OK** |
| End1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: End1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: End1 on line 1 | **OK** |
| EndIf | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: EndIfon line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: EndIfon line 1 | **OK** |
| EndIf˽ | Token = EndIf | Token = EndIf | **OK** |
| EndIf↵ | Token = EndIf | Token = EndIf | **OK** |
| EndIf: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: : on line 1 | **OK** |
| EndIf. | Lexical Analyzer Encountered 1 error/s.  Token = EndIfInvalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = EndIfInvalid input: . on line 1 | **OK** |
| EndIf< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: < on line 1 | **OK** |
| EndIf( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: ( on line 1 | **OK** |
| EndIf? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: ? on line 1 | **OK** |
| EndIf- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: - on line 1 | **OK** |
| EndIf+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: + on line 1 | **OK** |
| EndIf, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: , on line 1 | **OK** |
| EndIf| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: | on line 1 | **OK** |
| EndIf[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: [ on line 1 | **OK** |
| EndIf; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: ; on line 1 | **OK** |
| EndIf/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: EndIfon line 1  Invalid input: / on line 1 | **OK** |
| EndIff | Lexical Analyzer Encountered 1 error/s.  Invalid input: EndIff on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: EndIff on line 1 | **OK** |
| EndIf1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: EndIf1on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: EndIf1on line 1 | **OK** |
| For | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: For on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: For on line 1 | **OK** |
| For˽ | Lexical Analyzer Encountered 1 error/s.  Token = For | Lexical Analyzer Encountered 1 error/s.  Token = For | **OK** |
| For↵ | Token = For | Token = For | **OK** |
| For( | Lexical Analyzer Encountered 1 error/s.  Token = For  Invalid input: ( on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = For  Invalid input: ( on line 1 | **OK** |
| For: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: : on line 1 | **OK** |
| For. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: . on line 1 | **OK** |
| For? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: ? on line 1 | **OK** |
| For/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: / on line 1 | **OK** |
| For< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: < on line 1 | **OK** |
| For+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: + on line 1 | **OK** |
| For, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: , on line 1 | **OK** |
| For| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: | on line 1 | **OK** |
| For[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: [ on line 1 | **OK** |
| For; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: For on line 1  Invalid input: ; on line 1 | **OK** |
| Forr | Lexical Analyzer Encountered 1 error/s.  Invalid input: Forr on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Forr on line 1 | **OK** |
| For1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: For1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: For1 on line 1 | **OK** |
| If | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: If on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: If on line 1 | **OK** |
| If˽ | Token = If | Token = If | **OK** |
| If↵ | Token = If | Token = If | **OK** |
| If( | Lexical Analyzer Encountered 1 error/s.  Token = If  Invalid input: ( on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = If  Invalid input: ( on line 1 | **OK** |
| If: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: : on line 1 | **OK** |
| If. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: . on line 1 | **OK** |
| If? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: ? on line 1 | **OK** |
| If/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: / on line 1 | **OK** |
| If< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: < on line 1 | **OK** |
| If+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: + on line 1 | **OK** |
| If, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: , on line 1 | **OK** |
| If| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: | on line 1 | **OK** |
| If[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: [ on line 1 | **OK** |
| If; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: If on line 1  Invalid input: ; on line 1 | **OK** |
| Iff | Lexical Analyzer Encountered 1 error/s.  Invalid input: Iff on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Iff on line 1 | **OK** |
| If1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: If1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: If1 on line 1 | **OK** |
| Int | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Int on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Int on line 1 | **OK** |
| Int˽ | Token = Int | Token = Int | **OK** |
| Int↵ | Token = Int | Token = Int | **OK** |
| Int: | Lexical Analyzer Encountered 1 error/s.  Token = Int  Invalid input: < on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Int  Invalid input: < on line 1 | **OK** |
| Int. | Lexical Analyzer Encountered 1 error/s.  Token = Int  Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Int  Invalid input: . on line 1 | **OK** |
| Int< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: < on line 1 | **OK** |
| Int( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: ( on line 1 | **OK** |
| Int? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: ? on line 1 | **OK** |
| Int- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: - on line 1 | **OK** |
| Int+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: + on line 1 | **OK** |
| Int, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: , on line 1 | **OK** |
| Int| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: | on line 1 | **OK** |
| Int[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: [ on line 1 | **OK** |
| Int; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: ; on line 1 | **OK** |
| Int/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Int on line 1  Invalid input: / on line 1 | **OK** |
| Intt | Lexical Analyzer Encountered 1 error/s.  Invalid input: Intt on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Intt on line 1 | **OK** |
| Int1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Int1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Int1 on line 1 | **OK** |
| is | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: is on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: is on line 1 | **OK** |
| is˽ | Token = is | Token = is | **OK** |
| is↵ | Token = is | Token = is | **OK** |
| is: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: : on line 1 | **OK** |
| is. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: . on line 1 | **OK** |
| is( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: ( on line 1 | **OK** |
| is? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: ? on line 1 | **OK** |
| is- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: - on line 1 | **OK** |
| is+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: + on line 1 | **OK** |
| is, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: , on line 1 | **OK** |
| is| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: | on line 1 | **OK** |
| is[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: [ on line 1 | **OK** |
| is; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: is on line 1  Invalid input: ; on line 1 | **OK** |
| iss | Lexical Analyzer Encountered 1 error/s.  Invalid input: :iss line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: :iss line 1 | **OK** |
| is1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: : is1 line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: : is1 line 1 | **OK** |
| Lead | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Lead on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Lead on line 1 | **OK** |
| Lead˽ | Token = Lead | Token = Lead | **OK** |
| Lead↵ | Token = Lead | Token = Lead | **OK** |
| Lead: | Lexical Analyzer Encountered 1 error/s.  Token = Lead  Invalid input: : on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Lead  Invalid input: : on line 1 | **OK** |
| Lead. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: . on line 1 | **OK** |
| Lead( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: ( on line 1 | **OK** |
| Lead? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: ? on line 1 | **OK** |
| Lead- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: - on line 1 | **OK** |
| Lead+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: + on line 1 | **OK** |
| Lead, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: , on line 1 | **OK** |
| Lead| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: | on line 1 | **OK** |
| Lead[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: [ on line 1 | **OK** |
| Lead; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: ; on line 1 | **OK** |
| Lead< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Lead on line 1  Invalid input: < on line 1 | **OK** |
| Leadd | Lexical Analyzer Encountered 1 error/s.  Invalid input: Leadd on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Leadd on line 1 | **OK** |
| Lead1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Lead1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Lead1 on line 1 | **OK** |
| Let | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Let on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Let on line 1 | **OK** |
| Let˽ | Token = Let | Token = Let | **OK** |
| Let↵ | Token = Let | Token = Let | **OK** |
| Let: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: : on line 1 | **OK** |
| Let. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: . on line 1 | **OK** |
| Let( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: ( on line 1 | **OK** |
| Let? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: ? on line 1 | **OK** |
| Let- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: - on line 1 | **OK** |
| Let+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: + on line 1 | **OK** |
| Let, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: , on line 1 | **OK** |
| Let| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: | on line 1 | **OK** |
| Let[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: [ on line 1 | **OK** |
| Let; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Let on line 1  Invalid input: ; on line 1 | **OK** |
| Lett | Lexical Analyzer Encountered 1 error/s.  Invalid input: Lett on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Lett on line 1 | **OK** |
| Let1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Let1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Let1 on line 1 | **OK** |
| Loop | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Loop on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Loopon line 1 | **OK** |
| Loop˽ | Token = Loop | Token = Loop | **OK** |
| Loop↵ | Token = Loop | Token = Loop | **OK** |
| Loop: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loopon line 1  Invalid input: : on line 1 | **OK** |
| Loop. | Lexical Analyzer Encountered 1 error/s.  Token = Loop Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Loop Invalid input: . on line 1 | **OK** |
| Loop< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: < on line 1 | **OK** |
| Loop( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: ( on line 1 | **OK** |
| Loop? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: ? on line 1 | **OK** |
| Loop- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: Loop on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: Loop on line 1 | **OK** |
| Loop+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: + on line 1 | **OK** |
| Loop, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: , on line 1 | **OK** |
| Loop| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: | on line 1 | **OK** |
| Loop[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: [ on line 1 | **OK** |
| Loop; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: ; on line 1 | **OK** |
| Loop/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Loop on line 1  Invalid input: / on line 1 | **OK** |
| Loopp | Lexical Analyzer Encountered 1 error/s.  Invalid input: Loopp on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Loopp on line 1 | **OK** |
| Loop1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Loop1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Loop1 on line 1 | **OK** |
| LoopIf | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: LoopIf on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: LoopIf on line 1 | **OK** |
| LoopIf˽ | Token = LoopIf | Token = LoopIf | **OK** |
| LoopIf↵ | Token = LoopIf | Token = LoopIf | **OK** |
| LoopIf: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: : on line 1 | **OK** |
| LoopIf. | Token = LoopIf | Token = LoopIf | **OK** |
| LoopIf< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: < on line 1 | **OK** |
| LoopIf( | Lexical Analyzer Encountered 1 error/s.  Token = LoopIf  Invalid input: ( on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = LoopIf  Invalid input: ( on line 1 | **OK** |
| LoopIf? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: ? on line 1 | **OK** |
| LoopIf- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: - on line 1 | **OK** |
| LoopIf+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: + on line 1 | **OK** |
| LoopIf, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: , on line 1 | **OK** |
| LoopIf| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: | on line 1 | **OK** |
| LoopIf[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: [ on line 1 | **OK** |
| LoopIf; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: ; on line 1 | **OK** |
| LoopIf/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: LoopIf on line 1  Invalid input: / on line 1 | **OK** |
| LoopIff | Lexical Analyzer Encountered 1 error/s.  Invalid input: LoopIff on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: LoopIff on line 1 | **OK** |
| LoopIf1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: LoopIf1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: LoopIf1 on line 1 | **OK** |
| No | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: No on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: No on line 1 | **OK** |
| No˽ | Token = boollit | Token = boollit | **OK** |
| No↵ | Token = boollit | Token = boollit | **OK** |
| No. | Lexical Analyzer Encountered 1 error/s.  Token = No  Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = No  Invalid input: . on line 1 | **OK** |
| No; | Lexical Analyzer Encountered 1 error/s.  Token = No  Invalid input: ; on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = No  Invalid input: ; on line 1 | **OK** |
| No& | Lexical Analyzer Encountered 1 error/s.  Token = No  Invalid input: & on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = No  Invalid input: & on line 1 | **OK** |
| No| | Lexical Analyzer Encountered 1 error/s.  Token = No  Invalid input: | on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = No  Invalid input: | on line 1 | **OK** |
| No) | Lexical Analyzer Encountered 1 error/s.  Token = No  Invalid input: ) on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = No  Invalid input: ) on line 1 | **OK** |
| No( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: ( online 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: ( online 1 | **OK** |
| No? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: ? on line 1 | **OK** |
| No: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: : on line 1 | **OK** |
| No< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: < on line 1 | **OK** |
| No- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: - on line 1 | **OK** |
| No+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: + on line 1 | **OK** |
| No, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: , on line 1 | **OK** |
| No[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: No on line 1  Invalid input: [ on line 1 | **OK** |
| Noo | Lexical Analyzer Encountered 1 error/s.  Invalid input: Noo on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Noo on line 1 | **OK** |
| No1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: No1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: No1 on line 1 | **OK** |
| Null | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Null on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Null on line 1 | **OK** |
| Null˽ | Token = Null | Token = Null | **OK** |
| Null↵ | Token = Null | Token = Null | **OK** |
| Null: | Lexical Analyzer Encountered 1 error/s.  Token = Null  Invalid input: : on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Null  Invalid input: : on line 1 | **OK** |
| Null. | Lexical Analyzer Encountered 1 error/s.  Token = Null  Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Null  Invalid input: . on line 1 | **OK** |
| Null< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: < on line 1 | **OK** |
| Null( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: ( on line 1 | **OK** |
| Null? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: ? on line 1 | **OK** |
| Null- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: - on line 1 | **OK** |
| Null+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: + on line 1 | **OK** |
| Null, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: , on line 1 | **OK** |
| Null| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: | on line 1 | **OK** |
| Null[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: [ on line 1 | **OK** |
| Null; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: ; on line 1 | **OK** |
| Null/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Null on line 1  Invalid input: / on line 1 | **OK** |
| Nulll | Lexical Analyzer Encountered 1 error/s.  Invalid input: Nulll on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Nulll on line 1 | **OK** |
| Null1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Null1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Null1 on line 1 | **OK** |
| Object | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Object on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Object on line 1 | **OK** |
| Object˽ | Token = Object | Token = Object | **OK** |
| Object↵ | Token = Object | Token = Object | **OK** |
| Object: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: : on line 1 | **OK** |
| Object. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: . on line 1 | **OK** |
| Object( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: ( on line 1 | **OK** |
| Object? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: ? on line 1 | **OK** |
| Object- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: - on line 1 | **OK** |
| Object+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: + on line 1 | **OK** |
| Object, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: , on line 1 | **OK** |
| Object| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: | on line 1 | **OK** |
| Object[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: [ on line 1 | **OK** |
| Object; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Object on line 1  Invalid input: ; on line 1 | **OK** |
| Objectt | Lexical Analyzer Encountered 1 error/s.  Invalid input: Objectt on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Objectt on line 1 | **OK** |
| Object1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Object1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Object1 on line 1 | **OK** |
| Option | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Option on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Option on line 1 | **OK** |
| Option˽ | Token = Option | Token = Option | **OK** |
| Option↵ | Token = Option | Token = Option | **OK** |
| Option: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: : on line 1 | **OK** |
| Option. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: . on line 1 | **OK** |
| Option( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: ( on line 1 | **OK** |
| Option? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: ? on line 1 | **OK** |
| Option- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: - on line 1 | **OK** |
| Option+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: + on line 1 | **OK** |
| Option, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: , on line 1 | **OK** |
| Option| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: | on line 1 | **OK** |
| Option[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: [ on line 1 | **OK** |
| Option; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Option on line 1  Invalid input: ; on line 1 | **OK** |
| Optionn | Lexical Analyzer Encountered 1 error/s.  Invalid input: Optionn on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Optionn on line 1 | **OK** |
| Option1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Option1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Option1 on line 1 | **OK** |
| Or | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Or on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Or on line 1 | **OK** |
| Or˽ | Token = Or | Token = Or | **OK** |
| Or↵ | Token = Or | Token = Or | **OK** |
| Or: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: : on line 1 | **OK** |
| Or. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: . on line 1 | **OK** |
| Or( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: ( on line 1 | **OK** |
| Or? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: ? on line 1 | **OK** |
| Or- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: - on line 1 | **OK** |
| Or+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: + on line 1 | **OK** |
| Or, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: , on line 1 | **OK** |
| Or| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: | on line 1 | **OK** |
| Or[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: [ on line 1 | **OK** |
| Or; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Or on line 1  Invalid input: ; on line 1 | **OK** |
| Orr | Lexical Analyzer Encountered error/s.  Invalid input: : Orr line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: : Orr line 1 | **OK** |
| Or1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: : Or1 line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: : Or1 line 1 | **OK** |
| Otherwise | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Otherwise on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Otherwise on line 1 | **OK** |
| Otherwise˽ | Token = Otherwise | Token = Otherwise | **OK** |
| Otherwise↵ | Token = Otherwise | Token = Otherwise | **OK** |
| Otherwise: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: : on line 1 | **OK** |
| Otherwise. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: . on line 1 | **OK** |
| Otherwise( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: ( on line 1 | **OK** |
| Otherwise? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: ? on line 1 | **OK** |
| Otherwise- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: - on line 1 | **OK** |
| Otherwise+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: + on line 1 | **OK** |
| Otherwise, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: , on line 1 | **OK** |
| Otherwise| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: | on line 1 | **OK** |
| Otherwise[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: [ on line 1 | **OK** |
| Otherwise; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Otherwise on line 1  Invalid input: ; on line 1 | **OK** |
| Otherwisee | Lexical Analyzer Encountered 1 error/s.  Invalid input: Otherwisee on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Otherwisee on line 1 | **OK** |
| Otherwise1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Otherwise1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Otherwise1 on line 1 | **OK** |
| of | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: of on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: of on line 1 | **OK** |
| of˽ | Token = of | Token = of | **OK** |
| of↵ | Token = of | Token = of | **OK** |
| of: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: : on line 1 | **OK** |
| of. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: . on line 1 | **OK** |
| of( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: ( on line 1 | **OK** |
| of? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: ? on line 1 | **OK** |
| of- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: - on line 1 | **OK** |
| of+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: + on line 1 | **OK** |
| of, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: , on line 1 | **OK** |
| of| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: | on line 1 | **OK** |
| of[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: [ on line 1 | **OK** |
| Of; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: of on line 1  Invalid input: ; on line 1 | **OK** |
| off | Lexical Analyzer Encountered 1 error/s.  Invalid input: off on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: off on line 1 | **OK** |
| of1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: of1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: of1 on line 1 | **OK** |
| Read | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Read on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Read on line 1 | **OK** |
| Read˽ | Token = Read | Token = Read | **OK** |
| Read↵ | Token = Read | Token = Read | **OK** |
| Read: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: : on line 1 | **OK** |
| Read. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: . on line 1 | **OK** |
| Read( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: ( on line 1 | **OK** |
| Read? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: ? on line 1 | **OK** |
| Read- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: - on line 1 | **OK** |
| Read+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: + on line 1 | **OK** |
| Read, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: , on line 1 | **OK** |
| Read| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: | on line 1 | **OK** |
| Read[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: [ on line 1 | **OK** |
| Read; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Read on line 1  Invalid input: ; on line 1 | **OK** |
| Readd | Lexical Analyzer Encountered 1 error/s.  Invalid input: Readd on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Readd on line 1 | **OK** |
| Read1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Read1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Read1 on line 1 | **OK** |
| Response | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Response on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Response on line 1 | **OK** |
| Response˽ | Token = Response | Token = Response | **OK** |
| Response↵ | Token = Response | Token = Response | **OK** |
| Response: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: : on line 1 | **OK** |
| Response. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: . on line 1 | **OK** |
| Response( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: ( on line 1 | **OK** |
| Response? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: ? on line 1 | **OK** |
| Response- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: - on line 1 | **OK** |
| Response+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: + on line 1 | **OK** |
| Response, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: , on line 1 | **OK** |
| Response| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: | on line 1 | **OK** |
| Response[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: [ on line 1 | **OK** |
| Response; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Response on line 1  Invalid input: ; on line 1 | **OK** |
| Responsee | Lexical Analyzer Encountered 1 error/s.  Invalid input: Responsee on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Responsee on line 1 | **OK** |
| Response1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Response1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Response1 on line 1 | **OK** |
| Say | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Say on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Say on line 1 | **OK** |
| Say˽ | Token = Say | Token = Say | **OK** |
| Say↵ | Token = Say | Token = Say | **OK** |
| Say: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: : on line 1 | **OK** |
| Say. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: . on line 1 | **OK** |
| Say( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: ( on line 1 | **OK** |
| Say? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: ? on line 1 | **OK** |
| Say- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: - on line 1 | **OK** |
| Say+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: + on line 1 | **OK** |
| Say, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: , on line 1 | **OK** |
| Say| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: | on line 1 | **OK** |
| Say[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: [ on line 1 | **OK** |
| Say; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Say on line 1  Invalid input: ; on line 1 | **OK** |
| Sayy | Lexical Analyzer Encountered 1 error/s.  Invalid input: Sayy on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Sayy on line 1 | **OK** |
| Say1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Say1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Say1 on line 1 | **OK** |
| Skip | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Skip on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Skip on line 1 | **OK** |
| Skip˽ | Token = Skip | Token = Skip | **OK** |
| Skip↵ | Token = Skip | Token = Skip | **OK** |
| Skip. | Lexical Analyzer Encountered 1 error/s.  Token = Skip  Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Skip  Invalid input: . on line 1 | **OK** |
| Skip< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: < on line 1 | **OK** |
| Skip( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: ( on line 1 | **OK** |
| Skip? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: ? on line 1 | **OK** |
| Skip- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: - on line 1 | **OK** |
| Skip+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: + on line 1 | **OK** |
| Skip, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: , on line 1 | **OK** |
| Skip| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: | on line 1 | **OK** |
| Skip[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: [ on line 1 | **OK** |
| Skip; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: ; on line 1 | **OK** |
| Skip/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Skip on line 1  Invalid input: / on line 1 | **OK** |
| Skipp | Lexical Analyzer Encountered 2 error/s.  Invalid input: Skipp on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: Skipp on line 1 | **OK** |
| Skip1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Skip1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Skip1 on line 1 | **OK** |
| Start | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Start on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Start on line 1 | **OK** |
| Start˽ | Token = Start | Token = Start | **OK** |
| Start↵ | Token = Start | Token = Start | **OK** |
| Start: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: . on line 1 | **OK** |
| Start. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: . on line 1 | **OK** |
| Start( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: ( on line 1 | **OK** |
| Start? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: ? on line 1 | **OK** |
| Start- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: - on line 1 | **OK** |
| Start+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: + on line 1 | **OK** |
| Start, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: , on line 1 | **OK** |
| Start| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: | on line 1 | **OK** |
| Start[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: [ on line 1 | **OK** |
| Start; | Lexical Analyzer Encountered 2 error/s.  QProper delimiter expected: Start on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Start on line 1  Invalid input: ; on line 1 | **OK** |
| Startt | Lexical Analyzer Encountered 1 error/s.  Invalid input: Startt on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Startt on line 1 | **OK** |
| Start1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Start1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Start1 on line 1 | **OK** |
| State | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Start on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: State on line 1 | **OK** |
| State˽ | Token = State | Token = State | **OK** |
| State↵ | Token = State | Token = State | **OK** |
| State: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: : on line 1 | **OK** |
| State. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: . on line 1 | **OK** |
| State( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: ( on line 1 | **OK** |
| State? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: ? on line 1 | **OK** |
| State- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: - on line 1 | **OK** |
| State+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: + on line 1 | **OK** |
| State, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: , on line 1 | **OK** |
| State| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: | on line 1 | **OK** |
| State[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: [ on line 1 | **OK** |
| State; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: State on line 1  Invalid input: ; on line 1 | **OK** |
| Statee | Lexical Analyzer Encountered 1 error/s.  Invalid input: Statee on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Statee on line 1 | **OK** |
| State1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: State1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: State1 on line 1 | **OK** |
| Stop | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Stop on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Stop on line 1 | **OK** |
| Stop˽ | Token = Stop | Token = Stop | **OK** |
| Stop↵ | Token = Stop | Token = Stop | **OK** |
| Stop. | Lexical Analyzer Encountered 1 error/s.  Token = Stop  Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Stop  Invalid input: . on line 1 | **OK** |
| Stop< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: < on line 1 | **OK** |
| Stop( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: ( on line 1 | **OK** |
| Stop? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: ? on line 1 | **OK** |
| Stop- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: - on line 1 | **OK** |
| Stop+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: + on line 1 | **OK** |
| Stop, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: , on line 1 | **OK** |
| Stop| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: | on line 1 | **OK** |
| Stop[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: [ on line 1 | **OK** |
| Stop; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: ; on line 1 | **OK** |
| Stop/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Stop on line 1  Invalid input: / on line 1 | **OK** |
| Stopp | Lexical Analyzer Encountered 1 error/s.  Invalid input: Stopp on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Stopp on line 1 | **OK** |
| Stop1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Stop1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Stop1 on line 1 | **OK** |
| String | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: String on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: String on line 1 | **OK** |
| String˽ | Token = String | Token = String | **OK** |
| String↵ | Token = String | Token = String | **OK** |
| String: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: : on line 1 | **OK** |
| String. | Lexical Analyzer Encountered 1 error/s.  Token = String  Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = String  Invalid input: . on line 1 | **OK** |
| String< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: < on line 1 | **OK** |
| String( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: ( on line 1 | **OK** |
| String? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: ? on line 1 | **OK** |
| String- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: - on line 1 | **OK** |
| String+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: + on line 1 | **OK** |
| String, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: , on line 1 | **OK** |
| String| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: [ on line 1 | **OK** |
| String[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: [ on line 1 | **OK** |
| String; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: ; on line 1 | **OK** |
| String/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: String on line 1  Invalid input: / on line 1 | **OK** |
| Stringg | Lexical Analyzer Encountered 1 error/s.  Invalid input: Stringg on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Stringg on line 1 | **OK** |
| String1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: String1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: String1 on line 1 | **OK** |
| Task | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Task on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Task on line 1 | **OK** |
| Task˽ | Token = Task | Token = Task | **OK** |
| Task↵ | Token = Task | Token = Task | **OK** |
| Task: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: : on line 1 | **OK** |
| Task. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: . on line 1 | **OK** |
| Task( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: ( on line 1 | **OK** |
| Task? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: ? on line 1 | **OK** |
| Task- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: - on line 1 | **OK** |
| Task+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: + on line 1 | **OK** |
| Task, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: , on line 1 | **OK** |
| Task| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: | on line 1 | **OK** |
| Task[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: [ on line 1 | **OK** |
| Task; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Task on line 1  Invalid input: ; on line 1 | **OK** |
| Taskk | Lexical Analyzer Encountered 1 error/s.  Invalid input: Taskk on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Taskk on line 1 | **OK** |
| Task1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Task1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Task1 on line 1 | **OK** |
| Until | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Until on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Until on line 1 | **OK** |
| Until˽ | Token = Until | Token = Until | **OK** |
| Until↵ | Token = Until | Token = Until | **OK** |
| Until( | Lexical Analyzer Encountered 1 error/s.  Token = Until  Invalid input: ( on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Until  Invalid input: ( on line 1 | **OK** |
| Until: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: : on line 1 | **OK** |
| Until. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: . on line 1 | **OK** |
| Until? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: ? on line 1 | **OK** |
| Until- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: - on line 1 | **OK** |
| Until+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: + on line 1 | **OK** |
| Until, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: , on line 1 | **OK** |
| Until| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: | on line 1 | **OK** |
| Until[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: [ on line 1 | **OK** |
| Until; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: ; on line 1 | **OK** |
| Until/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Until on line 1  Invalid input: / on line 1 | **OK** |
| Untill | Lexical Analyzer Encountered 1 error/s.  Invalid input: Untill on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Untill on line 1 | **OK** |
| Until1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Until1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Until1 on line 1 | **OK** |
| Var | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Var on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Var on line 1 | **OK** |
| Var˽ | Token = Var | Token = Var | **OK** |
| Var↵ | Token = Var | Token = Var | **OK** |
| Var: | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: : on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: ; on line 1 | **OK** |
| Var. | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: . on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: . on line 1 | **OK** |
| Var( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: ( on line 1 | **OK** |
| Var? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: ? on line 1 | **OK** |
| Var- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: - on line 1 | **OK** |
| Var+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: + on line 1 | **OK** |
| Var, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: , on line 1 | **OK** |
| Var| | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: | on line 1 | **OK** |
| Var[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: [ on line 1 | **OK** |
| Var; | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: ; on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Var on line 1  Invalid input: ; on line 1 | **OK** |
| Varr | Lexical Analyzer Encountered 1 error/s.  Invalid input: Varr on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Varr on line 1 | **OK** |
| Var1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Var1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Var1 on line 1 | **OK** |
| Yes | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Yes on line 1 | Lexical Analyzer Encountered 1 error/s.  Proper delimiter expected: Yes on line 1 | **OK** |
| Yes˽ | Token = boollit | Token = boollit | **OK** |
| Yes↵ | Token = Yes | Token = Yes | **OK** |
| Yes. | Lexical Analyzer Encountered 1 error/s.  Token = Yes  Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Yes  Invalid input: . on line 1 | **OK** |
| Yes; | Lexical Analyzer Encountered 1 error/s.  Token = Yes  Invalid input: ; on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Yes  Invalid input: ; on line 1 | **OK** |
| Yes& | Lexical Analyzer Encountered 1 error/s.  Token = Yes  Invalid input: & on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Yes  Invalid input: & on line 1 | **OK** |
| Yes| | Lexical Analyzer Encountered 1 error/s.  Token = Yes  Invalid input: | on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Yes  Invalid input: | on line 1 | **OK** |
| Yes) | Lexical Analyzer Encountered 1 error/s.  Token = Yes  Invalid input: ) on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Yes  Invalid input: ) on line 1 | **OK** |
| Yes( | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: ( on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: ( on line 1 | **OK** |
| Yes? | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: ? on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: ? on line 1 | **OK** |
| Yes- | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: - on line 1 | **OK** |
| Yes+ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: + on line 1 | **OK** |
| Yes, | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: , on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: , on line 1 | **OK** |
| Yes[ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: [ on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: [ on line 1 | **OK** |
| Yes< | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: < on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: < on line 1 | **OK** |
| Yes/ | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: / on line 1 | Lexical Analyzer Encountered 2 error/s.  Proper delimiter expected: Yes on line 1  Invalid input: / on line 1 | **OK** |
| Yess | Lexical Analyzer Encountered 1 error/s.  Invalid input: Yess on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Yess on line 1 | **OK** |
| Yes1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Yes1 on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Yes1 on line 1 | **OK** |

***Identifier***

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **Actual Output** | **Remarks** |
| Num˽ | Token = id | Token = id | **OK** |
| AgE˽ | Token = id | Token = id | **OK** |
| CTR˽ | Token = id | Token = id | **OK** |
| aDD˽ | Lexical Analyzer Encountered 1 error/s.  Invalid input: aDD on line 1 | Lexical Analyzer Encountered 1 error/s.`  Invalid input: aDD on line 1 | **OK** |
| Num | Lexical Analyzer Encountered 1 error/s.  Invalid input: Num on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Num on line 1 | **OK** |
| 1age˽ | Lexical Analyzer Encountered 1 error/s.  Invalid input: 1age on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: 1age on line 1 | **OK** |
| 2Add˽ | Lexical Analyzer Encountered 1 error/s.  Invalid input: 2Add on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: 2Add on line 1 | **OK** |
| Ctr1˽ | Token = id | Token = id | **OK** |
| A200˽ | Token = id | Token = id | **OK** |
| N200um˽ | Token = id | Token = id | **OK** |
| Abcdefghijk˽ | Token = id | Token = id | **OK** |
| Nu\_m˽ | Token = id | Token = id | **OK** |
| Num\_ | Lexical Analyzer Encountered 1 error/s.  Invalid input: Num\_ on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: Num\_ on line 1 | **OK** |
| \_Num | Lexical Analyzer Encountered 1 error/s.  Token = id  Invalid input: \_ on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = id  Invalid input: \_ on line 1 | **OK** |
| Num\_˽ | Token = id | Token = id | **OK** |
| N\_u\_m˽ | Token = id | Token = id | **OK** |
| &um˽ | Lexical Analyzer Encountered 2 error/s.  Invalid input: & on line 1  Invalid input: um on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: & on line 1  Invalid input: um on line 1 | **OK** |
| N&m˽ | Lexical Analyzer Encountered 3 error/s.  Invalid input: N on line 1  Invalid input: & on line 1  Invalid input: m on line 1 | Lexical Analyzer Encountered 3 error/s.  Invalid input: N on line 1  Invalid input: & on line 1  Invalid input: m on line 1 | **OK** |
| Nu&˽ | Lexical Analyzer Encountered 2 error/s.  Invalid input: Nu on line 1  Invalid input: & on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: Nu on line 1  Invalid input: & on line 1 | **OK** |

***Literal***

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **Actual Output** | **Remarks** |
| **INT LITERAL** | | | |
| 1˽ | Token = intlit | Token = intlit | **OK** |
| 123˽ | Token = intlit | Token = intlit | **OK** |
| 1234567890˽ | Token = intlit | Token = intlit | **OK** |
| 0000000001˽ | Token = intlit | Token = intlit | **OK** |
| ~1˽ | Token = intlit | Token = intlit | **OK** |
| ~123˽ | Token = intlit | Token = intlit | **OK** |
| ~1234567890˽ | Token = intlit | Token = intlit | **OK** |
| ~0000000001˽ | Token = intlit | Token = intlit | **OK** |
| **DOUBLE LITERAL** | | | |
| 0.1˽ | Token = doublelit | Token = doublelit | **OK** |
| 1.12345˽ | Token = doublelit | Token = doublelit | **OK** |
| 12345.5˽ | Token = doublelit | Token = doublelit | **OK** |
| 12345.54321˽ | Token = doublelit | Token = doublelit | **OK** |
| 0.0˽ | Token = doublelit | Token = doublelit | **OK** |
| ~0.1˽ | Token = doublelit | Token = doublelit | **OK** |
| ~0.0˽ | Token = doublelit | Token = doublelit | **OK** |
| ~1.12345˽ | Token = doublelit | Token = doublelit | **OK** |
| ~12345.5˽ | Token = doublelit | Token = doublelit | **OK** |
| ~12345.54321˽ | Token = doublelit | Token = doublelit | **OK** |
| **CHAR LITERAL** | | | |
| ‘A’˽ | Token = charlit | Token = charlit | **OK** |
| ‘a’˽ | Token = charlit | Token = charlit | **OK** |
| ‘a˽ | Lexical Analyzer Encountered 1 error/s.  Invalid input: ‘a on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: ‘a on line 1 | **OK** |
| ‘a” ˽ | Lexical Analyzer Encountered 1 error/s.  Invalid input: ‘a” on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: ‘a” on line 1 | **OK** |
| ‘ABC‘˽ | Lexical Analyzer Encountered 1 error/s.  Invalid input: ‘ABC’ on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: ‘ABC’ on line 1 | **OK** |
| ‘1’˽ | Token = charlit | Token = charlit | **OK** |
| ‘&’ ˽ | Token = charlit | Token = charlit | **OK** |
| ‘;’ ˽ | Token = charlit | Token = charlit | **OK** |
| ‘+’ ˽ | Token = charlit | Token = charlit | **OK** |
| ‘[6]’ ˽ | Lexical Analyzer Encountered 1 error/s.  Invalid input: ‘[6]’ on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: ‘[6]’ on line 1 | **OK** |
| ‘?’ ˽ | Token = charlit | Token = charlit | **OK** |
| ‘.’ ˽ | Token = charlit | Token = charlit | **OK** |
| ‘’˽ | Token = charlit | Token = charlit | **OK** |
| ‘˽’˽ | Token = charlit | Token = charlit | **OK** |
| **STRING LITERAL** | | | |
| “Hello World” ˽ | Token = stringlit | Token = stringlit | **OK** |
| “hello world” ˽ | Token = stringlit | Token = stringlit | **OK** |
| “H” ˽ | Token = stringlit | Token = stringlit | **OK** |
| “h” ˽ | Token = stringlit | Token = stringlit | **OK** |
| “Hello!” ˽ | Token = stringlit | Token = stringlit | **OK** |
| “Hello?” ˽ | Token = stringlit | Token = stringlit | **OK** |
| “1” ˽ | Token = stringlit | Token = stringlit | **OK** |
| “123” ˽ | Token = stringlit | Token = stringlit | **OK** |
| “&”˽ | Token = stringlit | Token = stringlit | **OK** |
| “Hello World’˽ | Lexical Analyzer Encountered 1 error/s.  Invalid input: “Hello World’ on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: “Hello World’ on line 1 | **OK** |
| “”˽ | Token = stringlit | Token = stringlit | **OK** |
| “˽” ˽ | Token = stringlit | Token = stringlit | **OK** |
| “””˽ | Lexical Analyzer Encountered 2 error/s.  Invalid input: “” on line 1  Invalid input: “ on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: “” on line 1  Invalid input: “ on line 1 | **OK** |
| **BOOLEAN LITERALS** | | | |
| Boolean Num is Yes.˽ | Token = Boolean  Token = id  Token = is  Token = boollit  Token = . | Token = Boolean  Token = id  Token = is  Token = boollit  Token = . | **OK** |
| Boolean Num is No.˽ | Token = Boolean  Token = id  Token = is  Token = boollit  Token = . | Token = Boolean  Token = id  Token = is  Token = boollit  Token = . | **OK** |
| Boolean Num is yes.˽ | Lexical Analyzer Encountered 1 error/s.  Token = Boolean  Token = id  Token = is  Token = .  Invalid input: yes on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Boolean  Token = id  Token = is  Token = .  Invalid input: yes on line 1 | **OK** |
| Boolean Num is no.˽ | Lexical Analyzer Encountered 1 error/s.  Token = Boolean  Token = id  Token = is  Token = .  Invalid input: no on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = Boolean  Token = id  Token = is  Token = .  Invalid input: no on line 1 | **OK** |

***Reserved Symbols***

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **Actual Output** | **Remarks** |
| && | Lexical Analyzer Encountered 2 error/s.  Invalid input: & on line 1  Invalid input: & on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: & on line 1  Invalid input: & on line 1 | **OK** |
| &&˽ | Token = && | Token = && | **OK** |
| &&””˽ | Token = &&  Token = stringlit | Token = &&  Token = stringlit | **OK** |
| &&’’˽ | Token = &&  Token = charlit | Token = &&  Token = charlit | **OK** |
| &&(˽ | Token = &&  Token = ( | Token = &&  Token = ( | **OK** |
| &&!˽ | Token = &&  Token = ! | Token = &&  Token = ! | **OK** |
| &&1˽ | Token = &&  Token = intlit | Token = &&  Token = intlit | **OK** |
| &&A˽ | Token = &&  Token = id | Token = &&  Token = id | **OK** |
| &&+˽ | Lexical Analyzer Encountered 2 error/s.  Token = +  Invalid input: & on line 1  Invalid input: & on line 1 | Lexical Analyzer Encountered 2 error/s.  Token = +  Invalid input: & on line 1  Invalid input: & on line 1 | **OK** |
| &&&˽ | Lexical Analyzer Encountered 1 error/s.  Token = &&  Invalid input: & on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = &&  Invalid input: & on line 1 | **OK** |
| || | Lexical Analyzer Encountered 2 error/s.  Invalid input: | on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: | on line 1  Invalid input: | on line 1 | **OK** |
| ||˽ | Token = || | Token = || | **OK** |
| ||↵ | Token = || | Token = || | **OK** |
| ||””˽ | Token = ||  Token = stringlit | Token = ||  Token = stringlit | **OK** |
| ||’’˽ | Token = ||  Token = charlit | Token = ||  Token = charlit | **OK** |
| ||(˽ | Token = ||  Token = ( | Token = ||  Token = ( | **OK** |
| ||!˽ | Token = ||  Token = ! | Token = ||  Token = ! | **OK** |
| ||~1˽ | Token = ||  Token = intlit | Token = ||  Token = intlit | **OK** |
| ||1˽ | Token = ||  Token = intlit | Token = ||  Token = intlit | **OK** |
| ||A˽ | Token = ||  Token = id | Token = ||  Token = id | **OK** |
| ||+˽ | Lexical Analyzer Encountered 2 error/s.  Token = +  Invalid input: | on line 1  Invalid input: | on line 1 | Lexical Analyzer Encountered 2 error/s.  Token = +  Invalid input: | on line 1  Invalid input: | on line 1 | **OK** |
| |||˽ | Lexical Analyzer Encountered 1 error/s.  Token = ||  Invalid input: | on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = ||  Invalid input: | on line 1 | **OK** |
| . | Lexical Analyzer Encountered 1 error/s.  Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: . on line 1 | **OK** |
| . ˽ | Token = . | Token = . | **OK** |
| .↵ | Token = . | Token = . | **OK** |
| .#˽ | Token = .  Token = # | Token = .  Token = # | **OK** |
| .A˽ | Token = .  Token = id | Token = .  Token = id | **OK** |
|  |  |  |  |
|  |  |  |  |
| ..˽ | Lexical Analyzer Encountered 1 error/s.  Token = .  Invalid input: . on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = .  Invalid input: . on line 1 | **OK** |
| ! | Lexical Analyzer Encountered 1 error/s.  Invalid input: ! on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: ! on line 1 | **OK** |
| ! ˽ | Token = ! | Token = ! | **OK** |
| !↵ | Token = ! | Token = ! | **OK** |
| !””˽ | Token = ! Token = stringlit | Token = ! Token = stringlit | **OK** |
| !’’˽ | Token = ! Token = charlit | Token = ! Token = charlit | **OK** |
| !(˽ | Token = !  Token = ( | Token = !  Token = ( | **OK** |
| !~1˽ | Token = !  Token = intlit | Token = !  Token = intlit | **OK** |
| !1˽ | Token = !  Token = intlit | Token = !  Token = intlit | **OK** |
| !A˽ | Token = !  Token = id | Token = !  Token = id | **OK** |
| !!˽ | Token = !  Token = ! | Token = !  Token = ! | **OK** |
| != | Lexical Analyzer Encountered 2 error/s.  Invalid input: ! on line 1  Invalid input: = on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: ! on line 1  Invalid input: = on line 1 | **OK** |
| !=˽ | Token = != | Token = != | **OK** |
| !=↵ | Token = != | Token = != | **OK** |
| !=!˽ | Token = !=  Token = ! | Token = !=  Token = ! | **OK** |
| !=~1˽ | Token = !=  Token = intlit | Token = !=  Token = intlit | **OK** |
| !=1˽ | Token = !=  Token = intlit | Token = !=  Token = intlit | **OK** |
| !=A˽ | Token = !=  Token = id | Token = !=  Token = id | **OK** |
| @ | Lexical Analyzer Encountered 1 error/s.  Invalid input: @ on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: @ on line 1 | **OK** |
| @˽ | Lexical Analyzer Encountered 1 error/s.  Invalid input: @ on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: @ on line 1 | **OK** |
| @↵ | Lexical Analyzer Encountered 1 error/s.  Invalid input: @ on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: @ on line 1 | **OK** |
| @A˽ | Token = @  Token = id | Token = @  Token = id | **OK** |
| @&˽ | Lexical Analyzer Encountered 2 error/s.  Invalid input: @ on line 1  Invalid input: & on line 1 | Lexical Analyzer Encountered 2  error/s.  Invalid input: @ on line 1  Invalid input: & on line 1 | **OK** |
| @@˽ | Lexical Analyzer Encountered 2 error/s.  Invalid input: @ on line 1  Invalid input: @ on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: @ on line 1  Invalid input: @ on line 1 | **OK** |
| ( | Lexical Analyzer Encountered 1 error/s.  Invalid input: ( on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: ( on line 1 | **OK** |
| (˽ | Token = ( | Token = ( | **OK** |
| (↵ | Token = ( | Token = ( | **OK** |
| (!˽ | Token = (  Token = ! | Token = (  Token = ! | **OK** |
| (~1˽ | Token = (  Token = intlit | Token = (  Token = intlit | **OK** |
| (1˽ | Token = (  Token = intlit | Token = (  Token = intlit | **OK** |
| (A˽ | Token = (  Token = id | Token = (  Token = id | **OK** |
| (“”˽ | Token = (  Token = stringlit | Token = (  Token = stringlit | **OK** |
| (‘’˽ | Token = (  Token = charlit | Token = (  Token = charlit | **OK** |
| ((˽ | Token = (  Token = ( | Token = (  Token = ( | **OK** |
| )˽ | Token = ) | Token = ) | **OK** |
| )↵ | Token = ) | Token = ) | **OK** |
| )+˽ | Token = )  Token = + | Token = )  Token = + | **OK** |
| )-˽ | Token = )  Token = - | Token = )  Token = - | **OK** |
| )\*˽ | Token = )  Token = \* | Token = )  Token = \* | **OK** |
| )/˽ | Token = )  Token = / | Token = )  Token = / | **OK** |
| )%˽ | Token = )  Token = % | Token = )  Token = % | **OK** |
| ).˽ | Token = )  Token = . | Token = )  Token = . | **OK** |
| )1˽ | Token = )  Token = intlit | Token = )  Token = intlit | **OK** |
| ))˽ | Token = )  Token = ) | Token = )  Token = ) | **OK** |
| [ | Lexical Analyzer Encountered 1 error/s.  Invalid input: [ on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: [ on line 1 | **OK** |
| [˽ | Token = [ | Token = [ | **OK** |
| [↵ | Lexical Analyzer Encountered 1 error/s.  Invalid input: [ on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: [ on line 1 | **OK** |
| [1˽ | Token = [  Token = intlit | Token = [  Token = intlit | **OK** |
| [A˽ | Token = [  Token = id | Token = [  Token = ID | **OK** |
| [[˽ | Lexical Analyzer Encountered 1 error/s.  Token = [  Invalid input: [ on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = [  Invalid input: [ on line 1 | **OK** |
| ] | Lexical Analyzer Encountered 1 error/s.  Invalid input: ] on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: ] on line 1 | **OK** |
| ]˽ | Token = ] | Token = ] | **OK** |
| ]↵ | Token = ] | Token = ] | **OK** |
| ][˽ | Token = ]  Token = [ | Token = ]  Token = [ | **OK** |
| ]% | Token = ]  Token = % | Token = ]  Token = % | **OK** |
| ]+˽ | Token = ]  Token = + | Token = ]  Token = + | **OK** |
| ]-˽ | Token = ]  Token = - | Token = ]  Token = - | **OK** |
| ]\*˽ | Token = ]  Token = \* | Token = ]  Token = \* | **OK** |
| ]/˽ | Token = ]  Token = / | Token = ]  Token = / | **OK** |
| ].˽ | Token = ]  Token = . | Token = ]  Token = . | **OK** |
| ];˽ | Token = ]  Token = ; | Token = ]  Token = ; | **OK** |
| ])˽ | Token = ]  Token = ) | Token = ]  Token = ) | **OK** |
| ]]˽ | Lexical Analyzer Encountered 1 error/s.  Token = ]  Invalid input: ] on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = ]  Invalid input: ] on line 1 | **OK** |
| = | Lexical Analyzer Encountered 1 error/s.  Invalid input: = on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: = on line 1 | **OK** |
| =˽ | Token = = | Token = = | **OK** |
| =↵ | Token = = | Token = = | **OK** |
| =”˽ | Token = =  Token = “ | Token = =  Token = “ | **OK** |
| =’˽ | Token = =  Token = ‘ | Token = =  Token = ‘ | **OK** |
| =(˽ | Token = =  Token = ( | Token = =  Token = ( | **OK** |
| =!˽ | Token = =  Token = ! | Token = =  Token = ! | **OK** |
| =~1˽ | Token = =  Token = intlit | Token = =  Token = intlit | **OK** |
| =1˽ | Token = =  Token = intlit | Token = =  Token = intlit | **OK** |
| =A˽ | Token = =  Token = id | Token = =  Token = id | **OK** |
| == | Lexical Analyzer Encountered 2 error/s.  Invalid input: = on line 1  Invalid input: = on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: = on line 1  Invalid input: = on line 1 | **OK** |
| ==˽ | Token = == | Token = == | **OK** |
| ==↵ | Token = == | Token = == | **OK** |
| ==!˽ | Token = ==  Token = ! | Token = ==  Token = ! | **OK** |
| ==~1˽ | Token = ==  Token = intlit | Token = ==  Token = intlit | **OK** |
| ==1˽ | Token = ==  Token = intlit | Token = ==  Token = intlit | **OK** |
| ==A˽ | Token = ==  Token = id | Token = ==  Token = id | **OK** |
| ===˽ | Lexical Analyzer Encountered 1 error/s.  Token = ==  Invalid input: = on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = ==  Invalid input: = on line 1 | **OK** |
| > | Lexical Analyzer Encountered 1 error/s.  Invalid input: > on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: > on line 1 | **OK** |
| >˽ | Token = > | Token = > | **OK** |
| >↵ | Token = > | Token = > | **OK** |
| >!˽ | Token = >  Token = ! | Token = >  Token = ! | **OK** |
| >~1˽ | Token = >  Token = intlit | Token = >  Token = intlit | **OK** |
| >1˽ | Token = >  Token = intlit | Token = >  Token = intlit | **OK** |
| >A˽ | Token = >  Token = id | Token = >  Token = id | **OK** |
| >>˽ | Lexical Analyzer Encountered 1 error/s.  Token = >  Invalid input: > on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = >  Invalid input: > on line 1 | **OK** |
| >= | Lexical Analyzer Encountered 2 error/s.  Invalid input: > on line 1  Invalid input: = on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: > on line 1  Invalid input: = on line 1 | **OK** |
| >=˽ | Token = >= | Token = >= | **OK** |
| >=↵ | Token = >= | Token = >= | **OK** |
| >=!˽ | Token = >=  Token = ! | Token = >=  Token = ! | **OK** |
| >=~1˽ | Token = >=  Token = intlit | Token = >=  Token = intlit | **OK** |
| >=1˽ | Token = >=  Token = intlit | Token = >=  Token = intlit | **OK** |
| >=A˽ | Token = >=  Token = id | Token = >=  Token = id | **OK** |
| < | Lexical Analyzer Encountered 1 error/s.  Invalid input: < on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: < on line 1 | **OK** |
| <˽ | Token = < | Token = < | **OK** |
| <↵ | Token = < | Token = < | **OK** |
| <!˽ | Token = <  Token = ! | Token = <  Token = ! | **OK** |
| <~1˽ | Token = <  Token = intlit | Token = <  Token = intlit | **OK** |
| <1˽ | Token = <  Token = intlit | Token = <  Token = intlit | **OK** |
| <A˽ | Token = <  Token = id | Token = <  Token = id | **OK** |
| <<˽ | Lexical Analyzer Encountered 1 error/s.  Token = <  Invalid input: < on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = <  Invalid input: < on line 1 | **OK** |
| <= | Lexical Analyzer Encountered 2 error/s.  Invalid input: < on line 1  Invalid input: = on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: < on line 1  Invalid input: = on line 1 | **OK** |
| <=˽ | Token = <= | Token = <= | **OK** |
| <=↵ | Token = <= | Token = <= | **OK** |
| <=!˽ | Token = <=  Token = ! | Token = <=  Token = ! | **OK** |
| <=~1˽ | Token = <=  Token = intlit | Token = <=  Token = intlit | **OK** |
| <=1˽ | Token = <=  Token = intlit | Token = <=  Token = intlit | **OK** |
| <=A˽ | Token = <=  Token = id | Token = <=  Token = id | **OK** |
| + | Lexical Analyzer Encountered 1 error/s.  Invalid input: + on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: + on line 1 | **OK** |
| +˽ | Token = + | Token = + | **OK** |
| +↵ | Token = + | Token = + | **OK** |
| +!˽ | Token = +  Token = ! | Token = +  Token = ! | **OK** |
| +~1˽ | Token = +  Token = intlit | Token = +  Token = intlit | **OK** |
| +1˽ | Token = +  Token = intlit | Token = +  Token = intlit | **OK** |
| +A˽ | Token = +  Token = id | Token = +  Token = id | **OK** |
| +(˽ | Token = +  Token = ( | Token = +  Token = ( | **OK** |
| ++ | Lexical Analyzer Encountered 2 error/s.  Invalid input: + on line 1  Invalid input: + on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: + on line 1  Invalid input: + on line 1 | **OK** |
| ++˽ | Token = ++ | Token = ++ | **OK** |
| ++↵ | Token = ++ | Token = ++ | **OK** |
| ++)˽ | Token = ++  Token = ) | Token = ++  Token = ) | **OK** |
| ++.˽ | Token = ++  Token = . | Token = ++  Token = . | **OK** |
| ++,˽ | Token = ++  Token = , | Token = ++  Token = , | **OK** |
| ++;˽ | Token = ++  Token = ; | Token = ++  Token = ; | **OK** |
| ++A˽ | Token = ++  Token = id | Token = ++  Token = id | **OK** |
| +++˽ | Lexical Analyzer Encountered 1 error/s.  Token = ++  Invalid input: + on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = ++  Invalid input: + on line 1 | **OK** |
| - | Lexical Analyzer Encountered 1 error/s.  Invalid input: - on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: - on line 1 | **OK** |
| -˽ | Token = - | Token = - | **OK** |
| -↵ | Token = - | Token = - | **OK** |
| -!˽ | Token = -  Token = ! | Token = -  Token = ! | **OK** |
| -~1˽ | Token = -  Token = intlit | Token = -  Token = intlit | **OK** |
| -1˽ | Token = -  Token = intlint | Token = -  Token = intlint | **OK** |
| -A˽ | Token = -  Token = id | Token = -  Token = id | **OK** |
| -(˽ | Token = -  Token = ( | Token = -  Token = ( | **OK** |
| -- | Lexical Analyzer Encountered 2  error/s.  Invalid input: - on line 1  Invalid input: - on line 1 | Lexical Analyzer Encountered 2 error/s.  Invalid input: - on line 1  Invalid input: - on line 1 | **OK** |
| --˽ | Token = -- | Token = -- | **OK** |
| --↵ | Token = -- | Token = -- | **OK** |
| --)˽ | Token = --  Token = ) | Token = --  Token = ) | **OK** |
| --.˽ | Token = --  Token = . | Token = --  Token = . | **OK** |
| --,˽ | Token = --  Token = , | Token = --  Token = , | **OK** |
| --A˽ | Token = --  Token = id | Token = --  Token = id | **OK** |
| ---˽ | Lexical Analyzer Encountered 1 error/s.  Token = --  Invalid input: - on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = --  Invalid input: - on line 1 | **OK** |
| \* | Lexical Analyzer Encountered 1 error/s.  Invalid input: \* on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: \* on line 1 | **OK** |
| \*˽ | Token = \* | Token = \* | **OK** |
| \*↵ | Token = \* | Token = \* | **OK** |
| \*!˽ | Token = \*  Token = ! | Token = \*  Token = ! | **OK** |
| \*~1˽ | Token = \*  Token = intlit | Token = \*  Token = intlit | **OK** |
| \*1˽ | Token = \*  Token = intlit | Token = \*  Token = intlit | **OK** |
| \*A˽ | Token = \*  Token = id | Token = \*  Token = id | **OK** |
| \*(˽ | Token = \*  Token = ( | Token = \*  Token = ( | **OK** |
| \*=˽ | Lexical Analyzer Encountered 1 error/s.  Token = =  Invalid input: \* on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = =  Invalid input: \* on line 1 | **OK** |
| \*\*˽ | Lexical Analyzer Encountered 1 error/s.  Invalid input: \* on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: \* on line 1 | **OK** |
| / | Lexical Analyzer Encountered 1 error/s.  Invalid input: / on line | Lexical Analyzer Encountered 1 error/s.  Invalid input: / on line | **OK** |
| /˽ | Token = / | Token = / | **OK** |
| /↵ | Token = / | Token = / | **OK** |
| /!˽ | Token = / Invalid input: ! on line | Token = / Invalid input: ! on line | **OK** |
| /~1˽ | Token = /  Token = intlit | Token = /  Token = intlit | **OK** |
| /1˽ | Token = /  Token = intlit | Token = /  Token = intlit | **OK** |
| /A˽ | Token = /  Token = id | Token = /  Token = id | **OK** |
| /(˽ | Token = /  Token = ( | Token = /  Token = id | **OK** |
| /;˽ | Lexical Analyzer Encountered 1 error/s.  Invalid input: / on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: / on line 1 | **OK** |
| /=˽ | Lexical Analyzer Encountered 1 error/s.  Token = =  Invalid input: / on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = =  Invalid input: / on line 1 | **OK** |
| //˽ | Lexical Analyzer Encountered 1 error/s.  Token = /  Invalid input: / on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = /  Invalid input: / on line 1 | **OK** |
| ; | Lexical Analyzer Encountered 1 error/s.  Invalid input: ; on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: ; on line 1 | **OK** |
| ;˽ | Token = ; | Token = ; | **OK** |
| ;↵ | Token = ; | Token = ; | **OK** |
| ;””˽ | Token = ;  Token = stringlit | Token = ;  Token = stringlit | **OK** |
| ;’’˽ | Token = ;  Token = charlit | Token = ;  Token = charlit | **OK** |
| ;( ˽ | Token = ;  Token = ( | Token = ;  Token = ( | **OK** |
| ;! ˽ | Token = ;  Token = ! | Token = ;  Token = ! | **OK** |
| ;~1 ˽ | Token = ;  Token = intlit | Token = ;  Token = intlit | **OK** |
| ;1˽ | Token = ;  Token = intlit | Token = ;  Token = intlit | **OK** |
| ;A˽ | Token = ;  Token = stringlit | Token = ;  Token = stringlit | **OK** |
| ;;˽ | Lexical Analyzer Encountered 1 error/s.  Token = stringlit  Invalid input: ; on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = stringlit  Invalid input: ; on line 1 | **OK** |
| : | Lexical Analyzer Encountered 1 error/s.  Invalid input: : on line 1 | Lexical Analyzer Encountered 1 error/s.  Invalid input: : on line 1 | **OK** |
| :˽ | Token = : | Token = : | **OK** |
| :↵ | Token = : | Token = : | **OK** |
| :: ˽ | Lexical Analyzer Encountered 1 error/s.  Token = :  Invalid input: : on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = :  Invalid input: : on line 1 | **OK** |
| , ˽ | Token = , | Token = , | **OK** |
| ,↵ | Token = , | Token = , | **OK** |
| ,”” ˽ | Token = ,  Token = stringlit | Token = ,  Token = stringlit | **OK** |
| ,A˽ | Token = ,  Token = stringlit | Token = ,  Token = stringlit | **OK** |
| ,,˽ | Lexical Analyzer Encountered 1 error/s.  Token = ,  Invalid input: , on line 1 | Lexical Analyzer Encountered 1 error/s.  Token = ,  Invalid input: , on line 1 | **OK** |

***SYNTAX TEST SCRIPT***

**General Structure of the Program**Syntax: <program> → <global>Lead : Start <body> End . <taskdef>

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **Actual Output** | **Remarks** |
| Lead:  Start  Var Int Num is 11.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded... | **OK** |
| Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Int Num is 11. Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Task Int Age.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  If ((Num == 1) || (Num1 == 1))  Clear.  EndIf.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  If ((Num == 1) || (Num1 == 1))  Clear.  End.# | unexpected token "End", expected "EndIf", on line: 5 column: 1 | unexpected token "End", expected "EndIf", on line: 5 column: 1 | **OK** |
| Lead:  Start  End.  Task String Tree:  Start  Response “Hello”.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Var Int Num is 3  Clear.  End.# | Starting Syntax Analyzer  unexpected token "Clear", expected ".", on line: 4 column: 1 | Starting Syntax Analyzer  unexpected token "Clear", expected ".", on line: 4 column: 1 | **OK** |

**Constant Declaration**Syntax: <varlet> → Let <vardec>

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **ActualOutput** | **Remarks** |
| Let Int Age is 25.˽ Lead: Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Int Age is 35; Num is 45.**˽**  Lead: Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Int Age; Num is 55.  Lead: Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Int ; Num is 55.  Lead: Start  End.# | unexpected token ";", expected "id", on line: 1 column: 9 | unexpected token ";", expected "id", on line: 1 column: 9 | **OK** |
| Let Double Grade  is 1.5.  Lead: Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Double Grade is 1.5; Mile is 2.5.  Lead: Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Double ; Mile is 2.5.  Lead:  Start  End.# | unexpected token ";", expected "id", on line: 1 column: 12 | unexpected token ";", expected "id", on line: 1 column: 12 | **OK** |
| Let Char Letter is ‘A’.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Char Letter is ‘A’; Vowel is ‘E’.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Char Letter; Vowel is ‘E’.**˽**  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Char is 'A'.  Lead:  Start  End.# | unexpected token "is", expected "id", on line: 1 column: 10 | unexpected token "is", expected "id", on line: 1 column: 10 | **OK** |
| Let String Name is “Joe”.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let String Name is “Joe”; Surname is “Sam”.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let String Name; Surname is “Sam”.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let String is "Joe".  Lead:  Start  End.# | unexpected token "is", expected "id", on line: 1 column: 12 | unexpected token "is", expected "id", on line: 1 column: 12 | **OK** |
| Let Boolean Case is No.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Boolean Case is No; Status is Yes.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Boolean Case; Status is Yes.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Let Boolean No.  Lead:  Start  End.# | unexpected token "boollit", expected "id", on line: 1 column: 13 | unexpected token "boollit", expected "id", on line: 1 column: 13 | **OK** |

**Variable Declaration**Syntax: <varlet> → Var<vardec>

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **Actual Output** | **Remarks** |
| Var Int Age is 25.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Var Int Age is 35; Num is 45.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Var Int Age; Num is 55.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Var is 25.  End.# | unexpected token "is", expected "id", on line: 3 column: 9 | unexpected token "is", expected "id", on line: 3 column: 9 | **OK** |
| Var Double Grade  is 1.5.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Var Double Grade is 1.5; Mile is 2.5.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Var Double Grade; Mile is 2.5.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Var Double is 1.5.  End.# | unexpected token "is", expected "id", on line: 3 column: 12 | unexpected token "is", expected "id", on line: 3 column: 12 | **OK** |
| Var Char Letter is ‘A’.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Var Char Letter is ‘A’; Vowel is ‘E’.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Var Char Letter; Vowel is ‘E’.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Var Char is ‘A’.  End.# | unexpected token "is", expected "id", on line: 3 column: 10 | unexpected token "is", expected "id", on line: 3 column: 10 | **OK** |
| Var String Name is “Joe”.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Var String Name is “Joe”; Surname is “Sam”.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Var String Name; Surname is “Sam”.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Var String is “Joe”.  Lead:  Start  End.# | unexpected token "is", expected "id", on line: 3 column: 12 | unexpected token "is", expected "id", on line: 3 column: 12 | **OK** |
| Lead:  Start  Var Boolean Case is No.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Var Boolean Case is No; Status is Yes.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Var Boolean Case; Status is Yes.  Lead:  Start  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Var Boolean is No.  End.# | unexpected token "is", expected "id", on line: 3 column: 13 | unexpected token "is", expected "id", on line: 3 column: 13 | **OK** |

**Read Statement**Syntax: <input> → Read id

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **Actual Output** | **Remarks** |
| Lead:  Start  Read Age.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Read Age; Num.  End.# | unexpected token ";", expected ".", on line: 3 column: 9 | unexpected token ";", expected ".", on line: 3 column: 9 | **OK** |
| Lead:  Start  Read .  End.# | unexpected token ".", expected "id", on line: 3 column: 6 | unexpected token ".", expected "id", on line: 3 column: 6 | **OK** |
| Lead:  Start  Read Age@Year  End.# | unexpected token "@", expected ".", on line: 3 column: 9 | unexpected token "@", expected ".", on line: 3 column: 9 | **OK** |

**Say Statement**Syntax: <output> → Say <input\_statement>

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **Actual Output** | **Remarks** |
| Lead:  Start  Say “Hello”.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Say “Hello” , “World”.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Say ‘H’.  End.# | unexpected token "charlit", expected one of "stringlit", or "id", on line: 3 column: 5 | unexpected token "charlit", expected one of "stringlit", or "id", on line: 3 column: 5 | **OK** |
| Lead:  Start  Say Hello@World.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Say “Hello” , World@Wide.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Say .  End.# | unexpected token ".", expected one of "stringlit", or "id", on line: 3 column: 5 | unexpected token ".", expected one of "stringlit", or "id", on line: 3 column: 5 | **OK** |

**Assignment Statement**Syntax: <functions> → id <id\_choices>

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **Actual Output** | **Remarks** |
| Lead:  Start  Num = 5.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Test = 7.6.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  House = “The house is pink”.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Int Sum = (A + B).  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Int Subtract = (A – B).  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Int Multiply = (A \* B).  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Int Divide = (A / B).  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Int Modulus = (A % B).  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Fail = 5.0.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Choice = Yes.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  No = Decision.  End.# | unexpected token "boollit", expected "End", on line: 3 column: 1 | unexpected token "boollit", expected "End", on line: 3 column: 1 | **OK** |
| Lead:  Start  Day = ‘T’.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  X = Y.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  One = One.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  7= 2.  End.# | Invalid input: 7 on line 3 | Invalid input: 7 on line 3 | **OK** |
| Lead:  Start  ‘M’ = Month.  End.# | unexpected token "charlit", expected "End", on line: 3 column: 1 | unexpected token "charlit", expected "End", on line: 3 column: 1 | **OK** |
| Lead:  Start  Automata = “Subject”.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |

**Array Statement**Syntax: <array> → Array <dtype> id of <size>

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **Actual Output** | **Remarks** |
| Lead:  Start  Array Int Num of 3 by 2.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Array Int Num of Num1 by 2.  End.# | Expected: "intlit". | Expected: "intlit". | **OK** |
| Lead:  Start  Array Num of 1 by 2.  End.# | Expected: "Int", "Double", "Char", "String", "Double", "Boolean". | Expected: "Int", "Double", "Char", "String", "Double", "Boolean". | **OK** |
| Lead:  Start  Array Int of 1 by 2.  End.# | Expected: "id", . | Expected: "id", . | **OK** |
| Lead:  Start  Array Int Num 1 by 2.  End.# | Expected: "of", . | Expected: "of", . | **OK** |
| Lead:  Start  Array Int Num of by 2.  End.# | Expected: "intlit". | Expected: "intlit". | **OK** |
| Lead:  Start  Array Int Num of 1 2.  End.# | Expected: "by", ".". | Expected: "by", ".". | **OK** |
| Lead:  Start  Array Int Num of 1 by .  End.# | Expected: "intlit", . | Expected: "intlit", . | **OK** |
| Lead:  Start  Array Int Num of 1 by 2  End.# | Expected: ".", . | Expected: ".", . | **OK** |
| Lead:  Start  Int Num of 1 by 2.  End.# | Expected: "=", . | Expected: "=", . | **OK** |

**Conditional Statement**Syntax: <IfOtherwise> → If ( <conditions> ) <cond\_loop><or><otherwise>EndIf

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample Input** | **Expected Output** | **Actual Output** | **Remarks** |
| Lead:  Start  If ((Num1 == Num2) && (Num3 != Num4))  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  If ((Num1 Num2) && (Num3 != Num4))  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Expected: "@", "(", "&&", "||", ">=", "<=", "<", ">", "==", "!=", ")". | Expected: "@", "(", "&&", "||", ">=", "<=", "<", ">", "==", "!=", ")". | **OK** |
| Lead:  Start  If ((Num1 == Num2) && (Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Expected: "&&", "||", "==", "!=", ")". | Expected: "&&", "||", "==", "!=", ")". | **OK** |
| Lead:  Start  If ((Num1 == Num2) && (Num3 != Num4))  Say Num1, "and ", Num2, "is equals"  EndIf.  End.# | Expected: ",", ".". | Expected: ",", ".". | **OK** |
| Lead:  Start  ((Num1 == Num2) && (Num3 != Num4))  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Expected: "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option". | Expected: "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option". | **OK** |
| Lead:  Start  If ((Num1 == Num2) && (Num3 != Num4))  Say Num1, "and ", Num2, "is equals".  End.# | Expected: "Skip", "Stop", "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option", "Loop", "LoopIf", "Or", "intlit", "doublelit", "charlit", "stringlit", "boollit", "EndIf", "Otherwise". | Expected: "Skip", "Stop", "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option", "Loop", "LoopIf", "Or", "intlit", "doublelit", "charlit", "stringlit", "boollit", "EndIf", "Otherwise". | **OK** |
| Lead:  Start  If ((Num1 == Num2) (Num3 != Num4))  Say "Num1", "and ", Num2, "is equals".  EndIf.  End.# | Expected: "&&", "||", "==", "!=", ")". | Expected: "&&", "||", "==", "!=", ")". | **OK** |
| Lead:  Start  If (( == Num2) && (Num3 != Num4))  Say "Num1", "and ", Num2, "is equals".  EndIf.  End.# | Expected: "intlit", "doublelit", "charlit", "stringlit", "!", "id", "boollit", "(", "id". | Expected: "intlit", "doublelit", "charlit", "stringlit", "!", "id", "boollit", "(", "id". | **OK** |
| Lead:  Start  If ( == Num2) && (Num3 != Num4))  Say "Num1", "and ", Num2, "is equals".  EndIf.  End.# | Expected: "intlit", "doublelit", "charlit", "stringlit", "!", "id", "boollit", "(", "id". | Expected: "intlit", "doublelit", "charlit", "stringlit", "!", "id", "boollit", "(", "id". | **OK** |
| Lead:  Start  If Num2 == Num2) && (Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Expected: "(", . | Expected: "(", . | **OK** |
| Lead: Start If((Num1 == Num2) && Num3 != Num4) Say Num1 "and ", Num2, "is equals". EndIf. End.# | Expected: "@", ",", ".". | Expected: "@", ",", ".". | **OK** |
| Lead:  Start  If((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Or ((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Otherwise  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  If((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Or ((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Otherwise  Say Num1, "and ", Num2, "is equals"  EndIf.  End.# | Expected: ",", ".". | Expected: ",", ".". | **OK** |
| Lead:  Start  If((Num1 == Num2) && Num3 != Num4)  Say Num1 "and ", Num2, "is equals".  Or ((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Otherwise  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Expected: "@", ",", ".". | Expected: "@", ",", ".". | **OK** |
| Lead:  Start  If((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  ((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Otherwise  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Expected: "Skip", "Stop", "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option", "Loop", "LoopIf", "Or", "intlit", "doublelit", "charlit", "stringlit", "boollit", "EndIf", "Otherwise". | Expected: "Skip", "Stop", "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option", "Loop", "LoopIf", "Or", "intlit", "doublelit", "charlit", "stringlit", "boollit", "EndIf", "Otherwise". | **OK** |
| Lead:  Start  If((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  ((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Otherwise +  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Expected: "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option", "Loop", "LoopIf", "Skip", "Stop", "Or", "intlit", "doublelit", "charlit", "stringlit", "id", "boollit", "EndIf", "Otherwise". | Expected: "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option", "Loop", "LoopIf", "Skip", "Stop", "Or", "intlit", "doublelit", "charlit", "stringlit", "id", "boollit", "EndIf", "Otherwise". | **OK** |
| Lead:  Start  If((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Or == ((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Otherwise  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Expected: "(" | Expected: "(" | **OK** |
| Lead:  Start  If((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Or ((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Otherwise  Otherwise  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Expected: "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option", "Loop", "LoopIf", "Skip", "Stop", "Or", "intlit", "doublelit", "charlit", "stringlit", "id", "boollit", "EndIf", "Otherwise". | Expected: "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option", "Loop", "LoopIf", "Skip", "Stop", "Or", "intlit", "doublelit", "charlit", "stringlit", "id", "boollit", "EndIf", "Otherwise". | **OK** |
| Lead:  Start  If((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Or ((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Otherwise  EndIf.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  If((Num1 == Num2) && Num3 != Num4)  Or ((Num1 == Num2) && Num3 != Num4)  Otherwise  EndIf.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |
| Lead:  Start  Say Num1, "and ", Num2, "is equals".  Or ((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Otherwise  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Expected: "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option", ".", "Stop", "End". | Expected: "Var", "Clear", "id", "Int", "Char", "Boolean", "Double", "String", "++", "--", "Object", "Until", "Do", "For", "Array", "If", "Read", "Say", "Option", ".", "Stop", "End". | **OK** |
| Lead:  Start  If((Num1 == Num2) && Num3 != Num4@F  Say Num1, "and ", Num2, "is equals".  Or ((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Otherwise  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Expected: “@", "&&", "||", ">=", "<=", "<", ">", "==", "!=", ")". | Expected: “@", "&&", "||", ">=", "<=", "<", ">", "==", "!=", ")". | **OK** |
| Lead:  Start  If((Num1 == Num2) && Num3 != Num4@Float)  Say Num1, "and ", Num2, "is equals".  Or ((Num1 == Num2) && Num3 != Num4)  Say Num1, "and ", Num2, "is equals".  Otherwise  Say Num1, "and ", Num2, "is equals".  EndIf.  End.# | Syntax Analyzer Succeeded… | Syntax Analyzer Succeeded… | **OK** |