|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Q.1** | ex = 'NO  DO UNTIL ex = 'ÝES'     SAY 'LOOP 2 WAS ENTERED'       IF ex = 'YES ' THEN LEAVE     PULL ex  END | | |  |  |  | | --- | --- | --- | | 1. |  | The condition test is applied only at the bottom of the loop | | 2. |  | loop will always execute one time , even if the condition is false | | 3. |  | loop will always execute until the user enter YES | | 4. |  | all the above | | | |  |  |  |  |  | | --- | --- | | **Q.2** | Does REXX consider all these references to the same variable? | | |  |  |  | | --- | --- | --- | | 1. |  | the\_number | | 2. |  | THE\_NUMBER | | 3. |  | The\_Number | | 4. |  | all of the above | | | |  |  |  |  |  | | --- | --- | | **Q.3** | How does REXX view variable names internally | | |  |  |  | | --- | --- | --- | | 1. |  | lowercase | | 2. |  | uppercase | | 3. |  | mixedcase | | 4. |  | all of the above | | | |  |  | | | | | |
|  | |  | | |
| **Q.4** | | | Is the following  code correct  IF (var) THEN  SAY `VAR evaluated to 1'  ELSE SAY  `VAR evaluated to 0' | | |
| |  |  |  | | --- | --- | --- | | 1. |  | TRUE | | 2. |  | FALSE | | | | | | |
|  | | |  | | |
| **Q.5** | | What are three basic kind of parsing techniques? | | |
| |  |  |  | | --- | --- | --- | | 1. |  | by word | | 2. |  | by pattern | | 3. |  | by numeric pattern | | 4. |  | all the above | | | | | |

|  |  |
| --- | --- |
| **Q.6** | Whats the functionality of the Word-Oriented Function called "words"? |
| |  |  |  | | --- | --- | --- | | 1. |  | Returns the nth word in a string | | 2. |  | Retuns the number of words in a string | | 3. |  | Returns the last word of the given string | | 4. |  | Returns the first word of the given string | | |
|  |  |

|  |  |
| --- | --- |
| **Q.7** | Whats the functionality of the Word-Oriented Function called "words"? |
| |  |  |  | | --- | --- | --- | | 1. |  | Returns the nth word in a string | | 2. |  | Retuns the number of words in a string | | 3. |  | Returns the last word of the given string | | 4. |  | Returns the first word of the given string | | |
|  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Q.8** | | | | How the REXX perform the given statement-SAY x2b(c2x(xrange()) | | |
| |  |  |  | | --- | --- | --- | | 1. |  | first execute c2x, performs xrange() function and finally it runsx2b | | 2. |  | first performs the xrange() function then execute c2x, and finally it runsx2b | | 3. |  | first execute x2b,c2x and finally it run xrange() function | | 4. |  | first execute c2x,and x2b finally it performs xrange() function | | | | | | | |
| **Q.9** | | which of the following are the basis for arrays? | | |
| |  |  |  | | --- | --- | --- | | 1. |  | constant Symbols | | 2. |  | Compound Symbols | | 3. |  | No Symbols | | 4. |  | all of the above | | | | | |
|  | |  | | |
| **Q.10** | | Does REXX support Recursive function? |
| |  |  |  | | --- | --- | --- | | 1. |  | YES | | 2. |  | NO | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q.11** | | if any extra variable on parse arg will be set to  unitiailized. As always, an unitiailized variable is equal to its | | |
| |  |  |  | | --- | --- | --- | | 1. |  | own name in lowercase | | 2. |  | own name in uppercase | | 3. |  | both 1 and 2 | | 4. |  | None | | | | | |
| **Q.12** | | what is the scope of  the given variable in below  REXX exec? Calling\_rtn:     var\_1 = 'main'     var\_2 = 'main'     call my\_subrtn(var\_1)     say 'main:' var\_1 var-2    exit 0 my\_subrtn: procedure    arg var\_1    say 'my\_subrtn:' var\_1 var\_2    var\_1 = 'my\_subrtn'    var\_2 = 'my\_subrtn'    say 'my\_subrtn:' var\_1 var\_2    return |
| |  |  |  | | --- | --- | --- | | 1. |  | main var\_2 | | 2. |  | my\_subrtn my\_subrtn | | 3. |  | main main | | 4. |  | all the above | | | |
|  | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q.13** | | | which of the following one is REXX inconsistencies you'll have to remember? | |
| |  |  |  | | --- | --- | --- | | 1. |  | the arg function tells how manyarugments are passed into an internal routine | | 2. |  | applied to command-line arguments coming into a script, it always returns either 0 or 1 | | 3. |  | both 1 and 2 | | 4. |  | none | | | | | |
| **Q.14** | which of the following is persistent stream only, maintains independent positions? | |
| |  |  |  | | --- | --- | --- | | 1. |  | a read position | | 2. |  | a write position | | 3. |  | both 1 and 2 | | 4. |  | all of the above | | | |
|  |  | |

|  |  |
| --- | --- |
| **Q.15** | what are basic functions for standard character I/O? |
| |  |  |  | | --- | --- | --- | | 1. |  | charin | | 2. |  | charout | | 3. |  | output | | 4. |  | all of the above | | |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Q.16** | | | character I/O is |
| |  |  |  | | --- | --- | --- | | 1. |  | platform-dependent | | 2. |  | platform-independent | | 3. |  | both 1 and 2 | | 4. |  | none | | | | |
| **Q.17** | The three lines placed into the stack have all been removed, if we test the queue  at this point, what would be the return value of the queued() ? | | |
| |  |  |  | | --- | --- | --- | | 1. |  | 0 | | 2. |  | 1 | | 3. |  | 2 | | 4. |  | 3 | | | | |

|  |  |
| --- | --- |
| **Q.18** | first three lines of  data placed in the stack were retrieved in LIFO, then three more lines were placed into the stack and retrieve in FIFO order, which of the following is the correct order while retrieve the data from the stack? |
| |  |  |  | | --- | --- | --- | | 1. |  | LINE #3LINE #2LINE #1LINE #3LINE #2LINE #1 | | 2. |  | LINE #3LINE #2LINE #1LINE #1LINE #2LINE #3 | | 3. |  | LINE #3LINE #1LINE #2LINE #2LINE #1LINE #3 | | 4. |  | None | | |
|  |  |

|  |  |
| --- | --- |
| **Q.19** | which of the following choice can be included by developers of REXX interpreters |
| |  |  |  | | --- | --- | --- | | 1. |  | add a stack facility to the operating system | | 2. |  | create a stock "service" or "daemon" to provide this feature | | 3. |  | build stack into the interpreter itself | | 4. |  | all the above | | |
|  |  |

|  |  |
| --- | --- |
| **Q.20** | Queue stracture is refered to as |
| |  |  |  | | --- | --- | --- | | 1. |  | LIFO | | 2. |  | FIFO | | 3. |  | T | | 4. |  | all of the above | | |