Repetition Structure Practice Problems

# True/False Questions

1. Only WHILE loops can be nested.
2. Only FOR loops can be nested.
3. The main difference between a DO…WHILE and a WHILE loop is that a WHILE loop cannot result in an infinite loop.

# Multiple Choice Questions

1. This special input value is used to allow a user to end a loop:
   1. Stationary
   2. Sentinel
   3. Stop
   4. Iteration
2. The execution of one pass of a loop is referred to as a(n):
   1. Iteration
   2. Increment
   3. Initialization
   4. Individual
3. Repetition structures use what type of expression to determine if the block of code should run.
   1. Boolean
   2. Byte
   3. Bitlocker
   4. Bubble
4. Which of these is not a type of loop?
   1. FOR
   2. DO…WHILE
   3. WHILE…DO
   4. WHILE
5. This statement ends a loop immediately:
   1. HALT
   2. CONTINUE
   3. ESCAPE
   4. BREAK
6. This statement skips an iteration of a loop:
   1. NEXT
   2. CONTINUE
   3. SKIP
   4. BREAK
7. FOR Loops are often called what type of loop?
   1. Counting
   2. Correcting
   3. Incrementing
   4. Infinite

# Programming & Problem-Solving Questions – Reading Loops

What is the exact output of the following pseudocode segments?

1. FOR I from 0 to 3 by 1

FOR J from 1 to 4

PRINT((J+I) + "\t")

ENDFOR

PRINTLINE()

ENDFOR

1. FOR I from 0 to 2

FOR J from I\*3 to I\*3+3

PRINT(J + "; ")

ENDFOR

PRINTLINE()

ENDFOR

1. FOR I from 1 to 4

FOR J from 1 to 4

PRINT((J-I) + "\t")

ENDFOR

PRINTLINE()

ENDFOR

1. CREATE number = 67

DO

{

PRINT (number + “, “)

number /= 3

} WHILE (number >= 10)

1. int x = 5, y = 13;

for (int counter = 0; counter < 5; counter++)

{

PRINT("x = " + ++x + "\t");

PRINT("y = " + y++ + "\n");

}

1. int x = 2, y = 9;

for (int counter = 15; counter < 10; counter--)

{

PRINT("x = " + --x + "\t");

PRINT("y = " + y-- + "\n");

}

1. int x = 11, y = 3;

for (int counter = 10; counter > 5; counter--)

{

PRINT("x = " + x-- + "\t");

PRINT("y = " + --y + "\n");

}

1. int x = 3, y = 7;

for (int counter = 5; counter < 10; counter++)

{

PRINT("x = " + x++ + "\t");

PRINT("y = " + ++y + "\n");

}

1. int x = 2, y = 9;

for (int counter = 15; counter < 10; counter--)

{

PRINT("x = " + --x + "\t");

PRINT("y = " + y-- + "\n");

}

1. CREATE number = 1

DO

{

PRINT (number + “, “)

number += 3

} WHILE (number <= 5)

1. CREATE number = 17

DO

{

PRINT (number + “, “)

number \*= 3

} WHILE (number <= 100)

Print the exact output from the following pseudocode programs:

BEGIN MAIN

CREATE teamH = 0, teamV = 0, mercy = false

DO

teamH +=7

teamV += 3

IF (teamH >= (teamV + 20))

mercy = true

END IF

WHILE (mercy == false)

END WHILE

PRINTLINE(“Final Score: “ + teamH + “ to “ + teamV)

END MAIN

BEGIN MAIN

CREATE teamH = 0, teamV = 0, innings = 9

DO

IF (innings % 2 == 0)

teamH += 1

teamV += 2

ELSE

teamH += 2

teamV += 1

END IF

innings –

WHILE (innings > 0)

END WHILE

PRINTLINE(“Final Score: “ + teamH + “ to “ + teamV)

END MAIN

BEGIN MAIN

CREATE teamH = 0, teamV = 0, period = 0, penalty = false

DO

IF (penalty == false)

teamH++

teamV += 2

ELSE

teamH +=3

penalty = true

END IF

period ++

WHILE (period != 3)

END WHILE

PRINTLINE(“Final Score: “ + teamH + “ to “ + teamV)

END MAIN

BEGIN MAIN

CREATE timeLeft = 150, routines = 0

DO

IF (timeLeft > 120)

routines += 3

ELSE IF (timeLeft > 90)

routines += 2 ELSE IF (timeLeft > 0)

routines ++

END IF

timeLeft -= 20

WHILE (timeLeft >= 0)

END WHILE

PRINTLINE(“Final Number of Routines performed: “ + routines)

END MAIN

BEGIN MAIN

CREATE teamH = 0, teamV = 0, quarter = 1, control = ‘h’

DO

IF (control == ‘h’)

teamH += 2

teamV ++

control = ‘v’

ELSE

teamH ++

teamV += 3

control = ‘h’

END IF

quarter++

WHILE (quarter < 4)

END WHILE

PRINTLINE(“Final Score: “ + teamH + “ to “ + teamV)

END MAIN

BEGIN MAIN

CREATE teamH = 0, teamV = 0, holes = 0, penalty = false

DO

IF (penalty == false)

teamH –

teamV –

penalty = true

ELSE

teamH += 2

teamV ++

penalty = false

END IF

holes ++

WHILE (holes != 18)

END WHILE

PRINTLINE(“Final Score: “ + teamH + “ to “ + teamV)

END MAIN

BEGIN MAIN

CREATE teamH = 0, teamV = 0, gameOver = false, possession = ‘v’

DO

IF (possession == ‘v’)

teamH++

teamV += 2

possession = ‘h’

ELSE

teamH +=3

teamV ++

possession = ‘v’

END IF

IF (teamH > teamV)

gameOver = true

WHILE (!gameOver)

END WHILE

PRINTLINE(“Final Score: “ + teamH + “ to “ + teamV)

END MAIN

BEGIN MAIN

CREATE y = 1

FOR (CREATE count = 0, count < 6, count ++)

IF (count == 3)

CONTINUE

END IF

y = y + (count \* y)

PRINT y + “,”

END FOR

END MAIN

BEGIN MAIN

CREATE salary = 50000

CREATE INCREASE = 3000

CREATE year = 0

FOR (year = 1, year <= 5, year++)

PRINTLINE (“Year “ + year + “ Salary $” + salary) Salary += increase

END FOR

END MAIN

BEGIN MAIN

CREATE salary = 30000

CREATE INCREASE = 5000

CREATE year = 0

FOR (year = 1, year <= 5, year++)

PRINTLINE (“Year “ + year + “ Salary $” + salary)

Salary += increase

END FOR

END MAIN

BEGIN MAIN

CREATE salary = 25000

CREATE INCREASE = 7500

CREATE year = 0

FOR (year = 1, year <= 5, year++)

PRINTLINE (“Year “ + year + “ Salary $” + salary)

Salary += increase

END FOR

END MAIN

BEGIN MAIN

CREATE count = 0, outer, inner

FOR (outer = 0, outer <= 7, outer++)

FOR (inner = 10, inner >= 2, inner--)

count++;

END FOR

END FOR

PRINTLINE (“Count is: “ + count)

PRINTLINE (“Inner is: “ + inner)

PRINTLINE (“Outer is: “ + outer)

END MAIN

1. What is the final value of **count** after this code is executed?
2. What is the final value of **outer** after this code is executed?
3. What is the final value of **inner** after this code is executed?

BEGIN MAIN  
CREATE num1 = 15

CREATE num2 = 30

FOR each I from 1 to 4 by 1  
 IF (num1 == num2) THEN  
 PRINT(“lime” + ”\t”)

ENDIF  
 PRINT("lemon” + “\t”)  
 num1 = num1 \* 2  
 PRINTLINE("apple” + “\t”)

END FOR  
PRINTLINE(“orange”)

END MAIN

BEGIN MAIN  
CREATE num1 = 15

CREATE num2 = 60

FOR each I from 1 to 4 by 1  
 IF (num1 != num2) THEN  
 PRINT(“lime” + ”\t”)

ENDIF  
 PRINT("lemon” + “\t”)  
 num1 = num1 \* 4  
 PRINTLINE("apple” + “\t”)

END FOR  
PRINTLINE(“orange”)

END MAIN

BEGIN MAIN  
CREATE num1 = 60

CREATE num2 = 30

FOR each I from 1 to 4 by 1  
 IF (num1 <= num2) THEN  
 PRINT(“lime” + ”\t”)

ENDIF  
 PRINT("lemon” + “\t”)  
 num1 = num1 / 2  
 PRINTLINE("apple” + “\t”)

END FOR  
PRINTLINE(“orange”)

END MAIN

BEGIN MAIN  
CREATE num1 = 10

CREATE num2 = 30

FOR each I from 1 to 4 by 1  
 IF (num1 >= num2) THEN  
 PRINT(“lime” + ”\t”)

ENDIF  
 PRINT("lemon” + “\t”)  
 num1 = num1 \* 4  
 PRINTLINE("apple” + “\t”)

END FOR  
PRINTLINE(“orange”)

END MAIN

BEGIN MAIN  
CREATE num1 = 80

CREATE num2 = 10

FOR each I from 1 to 4 by 1  
 IF (num2 == num1) THEN  
 PRINT(“lime” + ”\t”)

ENDIF  
 PRINT("lemon” + “\t”)  
 num2 = num2 \* 2  
 PRINTLINE("apple” + “\t”)

END FOR  
PRINTLINE(“orange”)

END MAIN

BEGIN MAIN

 CREATE x = 1

 FOR (CREATE count = 0, count < 6, count++)

     IF (count == 3) CONTINUE

          x = x + (count \* x)

     PRINT(x + ",")

 END FOR

 PRINTLINE();

END MAIN

BEGIN MAIN

 CREATE x = 0

 FOR (CREATE count = 0, count < 10, count++)

  IF (count == 4) CONTINUE

      x += count

  END IF

  PRINT(x +  ", ")

 END FOR

 PRINTLINE()

END MAIN

BEGIN MAIN

 CREATE x = 0

 FOR (CREATE count = 0, count < 10, count++)

  x += count

  PRINT ( x + ", ")

  IF (count == 8) BREAK

 END FOR

 PRINTLINE()

END MAIN

BEGIN MAIN

CREATE A = 5, B = 10, sum = 0, count = 1

DO

sum = sum + count

count++

WHILE (count < A)

PRINTLINE (“Sum is: “ + sum)

END DO

END MAIN

BEGIN MAIN

CREATE A = 5, B = 10, sum = 0, count = 10

DO

sum = sum + count

count--

WHILE (count > B)

PRINTLINE (“Sum is: “ + sum)

END DO

END MAIN

BEGIN MAIN

CREATE A = 5, B = 10, sum = 0, count = 1

DO

sum = sum + count

count++

WHILE (count < B)

PRINTLINE (“Sum is: “ + sum)

END DO

END MAIN

BEGIN MAIN

CREATE A = 5, B = 10, sum = 0, count = 10

DO

sum = sum + count

count--

WHILE (count > A)

PRINTLINE (“Sum is: “ + sum)

END DO

END MAIN

# Programming & Problem-Solving Questions – Writing Loops

1) Write a loop that prints out the following sequence of numbers. The method must use a for-loop to print the outputs. HINT: “To get started: what’s the pattern from number X to (X+1)? Does it apply to the next pair of numbers?”

* 1. 3 7 13 21 31 43 57
  2. 8 12 18 26 36 48 62
  3. 5 10 15 20 25 30 35 40 45 50
  4. 63 56 49 42 35 28 21 14 7
  5. 15625 125 25 5
  6. 50 45 40 35 30 25 20 15 10 5
  7. 4 8 12 20 32 52 84 136
  8. 2 4 16 256 65536
  9. 80 72 64 56 48 40 32 24 16 8
  10. 7 14 21 28 35 42 49 56 63 70 77 84
  11. 1024 512 256 128 64 32 16
  12. 12 24 36 48 60 72 84
  13. 2 4 8 16 32 64 128
  14. 77 66 55 44 33 22 11
  15. 4 12 20 28 36 44 52
  16. 25 22 19 16 13 10 7 4 1
  17. 1 4 7 10 13 16 19 22 25
  18. 7 14 21 28 35 42 49 56 63

1. Write a **while loop** to sum all the values between 2 integers (A & B, input by the user), including A and B, and print the resulting sum. A must be less than B, otherwise print 0.
2. Mathematically, the Factorial of number N (denoted as N!) is the multiplication of all numbers between 1 and N. Write a **while** **loop** segment to implement Factorial of N.
3. Assume that the upper case alphabetic characters are ranked such that: ‘A’ < ‘B’ < ‘C’ < … < ’X’, < ‘Y’ < ‘Z’. Write a DO… WHILE loop that identifies the largest upper case alphabetic character entered by the user. You are not allowed to use arrays, lists, or any other data structure to store the entered characters. Your code must use a DO… WHILE loop, and must prompt (i.e. ask) the user to enter upper case alphabetic characters (i.e. ‘A’, ‘M’, ‘Z’, etc.) For each iteration of the loop, the user will be prompted for, and will enter only one character at a time. When the user is done entering characters, he/she will enter a ‘1’ to indicate no more input. Finally, after the DO… WHILE loop ends, your code should print a message indicating the largest alphabetic character entered by the user, i.e. “The largest alphabetic character entered was M.”

For example: Please enter a character: ‘K’, Please enter a character: ‘B’, Please enter a character: ‘N’, Please enter a character: ‘L’, Please enter a character: ‘1’. The message printed after the loop ends should be: “The largest alphabetic character entered was N.”

1. Write a DO… WHILE loop that identifies the largest integer entered by the user. You are not allowed to use arrays, lists, or any other data structure to store the entered values. Your code must use a DO… WHILE loop, and should prompt (i.e. ask) the user to enter positive integer values. For each iteration of the loop, the user will be prompted for, and will enter only one positive integer at a time. When the user is done entering numbers, he/she will enter a -1 to indicate no more input. Finally, after the DO… WHILE loop ends, your code should print a message indicating the largest value entered by the user, i.e. “The largest value entered was 23.”

For example: Please enter an integer: 24, Please enter an integer: 5, Please enter an integer: 102, Please enter an integer: 9, Please enter an integer: -1. The message printed after the loop ends should be: “The largest value entered was 102.”

1. Write a DO… WHILE loop that identifies the smallest number entered by the user. You are not allowed to use arrays, lists, or any other data structure to store the entered values. Your code must use a DO… WHILE loop, and should prompt (i.e. ask) the user to enter positive integer values. For each iteration of the loop, the user will be prompted for, and will enter only one positive integer at a time. When the user is done entering numbers, he/she will enter a -1 to indicate no more input. Finally, after the DO… WHILE loop ends, your code should print a message indicating the smallest value entered by the user, i.e. “The smallest value entered was 23.”

For example: Please enter an integer: 24, Please enter an integer: 5, Please enter an integer: 102, Please enter an integer: 9, Please enter an integer: -1. The message printed after the loop ends should be: “The smallest value entered was 5.”

1. Assume that the upper-case alphabetic characters are ranked such that: ‘A’ < ‘B’ < ‘C’ < … < ’X’, < ‘Y’ < ‘Z’. Write a DO… WHILE loop that identifies the smallest upper case alphabetic character entered by the user. You are not allowed to use arrays, lists, or any other data structure to store the entered characters. Your code must use a DO… WHILE loop, and must prompt (i.e. ask) the user to enter upper case alphabetic characters (i.e. ‘A’, ‘M’, ‘Z’, etc.) For each iteration of the loop, the user will be prompted for, and will enter only one character at a time. When the user is done entering characters, he/she will enter a ‘1’ to indicate no more input. Finally, after the DO… WHILE loop ends, your code should print a message indicating the smallest alphabetic character entered by the user, i.e. “The smallest alphabetic character entered was M.”

For example: Please enter a character: ‘K’, Please enter a character: ‘B’, Please enter a character: ‘N’, Please enter a character: ‘L’, Please enter a character: ‘1’. The message printed after the loop ends should be: “The smallest alphabetic character entered was B.”

1. One of the most frustrating things about technology is forgetting your password, incorrectly guessing it multiple times, and then being locked out of your account. Using either a WHILE or DO-WHILE loop (only), write a program that will repeatedly ask the user for his/her name and password, and then print “Welcome!” upon a successful login. However, after 10 incorrect attempts, they should not receive a greeting, but instead the message “See system administrator to reset password, sucker!” For this question, accept only one account name: “jbub” whose sickening password is “SparklingPuppies”.
2. Write a complete program that declares a variable and uses a loop to ask the student to enter the caloric value of each food item consumed in a day. The values should be greater than 0 (validate them using a loop) and the user should be able to enter a value to end the loop. The total amount of calories consumed in a day should be tracked and printed out in a statement indicating the student’s total intake for the day.
3. Write a complete program that declares a variable and uses a loop to ask the student to enter the caloric value of each food item consumed in a day. The values should be greater than 0 (validate them using a loop) and the user should be able to enter a value to end the loop. The highest number of calories consumed in a day should be tracked and printed out in a statement indicating the student’s highest caloric food item for the day.
4. Write a complete program that declares a variable of an appropriate type, asks for user input, stores the input in the variable and then uses a WHILE loop that ensures that the user has input a number that is negative and odd.
5. Write a complete program that declares a variable of an appropriate type, asks for user input, stores the input in the variable and then uses a WHILE loop that ensures that the user has input a number that is positive and even.
6. Write a complete program that declares a variable of an appropriate type, asks for user input, stores the input in the variable and then uses a WHILE loop that ensures that the user has input a number that is divisible by 3 and less than 10,000.
7. Given a non-zero positive integer n, use a DO-WHILE loop to calculate n! (n-factorial), where n! = n \* (n-1) \* (n-2) \* (n – 3) … \* (1). For example, if n = 4 then 4! = 4 \* 3 \* 2 \* 1 = 24. After the loop ends, print the value of n-factorial. Your code should declare and initialize any variables needed.
8. Given two non-zero positive integers x and y, use a DO-WHILE loop to calculate z, where z = x raised to the power of y. For example, if x = 3 and y = 4 then z = 3 \* 3 \* 3 \* 3 = 81. After the loop ends, print the value of z. Your code should declare and initialize any variables needed.
9. Given a non-zero positive integer n, write a DO-WHILE loop that calculates the product of all numbers from n to 1 that are multiples of 3. For example, if n is 10, we multiply the numbers 9, 6, 3 to get a product of 162. After the loop ends, print the product. Your code should declare and initialize any variables needed.
10. Using a do/while loop, write a program that continuously asks the user for two numbers and prints out the first number mod (%) the second number.  The program should stop when the result of the modulus is 0.
11. Using a do/while loop, write a code snippet (no main) that continues to ask users for a number until they enter -1.  Each pass through the loop, it should print out a running sum of numbers and the average of the numbers entered.  The -1 will be part of the sum and average.
12. Using a do/while loop, write a program that continuously asks the user to enter positive numbers until they type -1.  For each repetition of the loop, the largest value entered so far will be printed.