Name: Class:

Task 1

Harriet has written pseudocode for a program to calculate the total value of sales. Her pseudocode is shown below. Line numbers are given for reference, they are not part of the pseudocode.

1 total = 0

2 do

3 sales = int(input("Enter next sales value"))

4 //add sales to total

5 until sales == 0

6 print("Total:")

7 print(total)

(a) Harriet enters the numbers 25.00, 5.00, 6.50, 0.

What will be printed at Line 7?

(b) Line 4 is currently a comment. Complete line 4 in pseudocode.

(c) Draw a flow diagram that corresponds to this pseudocode.

|  |
| --- |
|  |

(d) Add lines to the pseudocode, above, so that it counts the number of sales and calculates and outputs the average sale.

(e) Test your program by tracing through it with the sales values given in part (a). Does it work correctly? If not, try and correct it.

(f) Does the program work whatever the user enters? If not, why not? Can you make sure it will not crash whatever the user enters?

Task 2

James is writing a program to simulate a dice game. The function Random(1,6) generates a number between 1 and 6. He has drawn a flowchart to represent the algorithm to calculate a player’s score when it is their turn. Paul and Coleen play the game.

score = score + die1   
+ die 2

die1 = random(1,6)

die2 = random(1,6)

Another go?

OUTPUT score

Start

End

score = 0

die1==die2?

No

Yes

Yes

No

(a) Describe the rules of the game.

(b) Paul rolls the dice three times, getting six and two on the first throw, one and four on the second throw and two and three on the third throw. Coleen also rolls the dice three times, getting five and six on the first throw, four and six on the second throw, two and two on the third throw.

What are the scores of each player?

(c) James has tried to rewrite the algorithm using pseudocode instead of a flowchart. Unfortunately his pseudocode does not work correctly.

Correct the code below by inserting one more statement.

do

die1 = random (1,6)

die2 = random (1,6)

if die1 == die2 then

score = 0

else

score = score + die1 + die2

print("score = " + str(score))

anotherGo = input("another go?")

endif

until anotherGo = "no"

print("score = " + str(score))

Task 3

The following program code is part of a game in which Mighty Max is engaged with the enemy, dealing deathly blows in all directions to overcome the evil gremlins but taking considerable punishment himself until he is fatally wounded.

wounds = 0

gremlins = 4

strength = 30

while strength > 0

if gremlins >= 1 then

wounds = wounds + gremlins

endif

if strength > 2 then

gremlins = gremlins - 1

print("Mighty Max has dealt a deathly blow to a gremlin, but his   
 strength is fading")

endif

strength = strength - wounds

print(wounds)  
 print(gremlins)   
 print(strength)

endwhile

print("Alas, our hero has been overcome… game over")

Complete the next 4 rows in the following trace table.

|  |  |  |  |
| --- | --- | --- | --- |
| **wounds** | **gremlins** | **strength** | **strength > 0** |
| 0 | 4 | 30 | True |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Task 4

When the creator of the game of chess showed his invention to the ruler of the country. The ruler was so pleased that he gave the inventor the right to name his prize for the invention. The man, who was very clever, asked the king for the following as his prize:

One grain of rice on the first square  
Two grains on the second square  
Four grains on the third square – and doubling for each square thereafter.

A programmer decides to find out how many grains of wheat the inventor will end up with. He tests the program with just 6 squares of the chessboard.

The program is shown below. Complete the trace table to show the total number of grains of wheat received for the first 6 squares of the chessboard.

print("How many grains of wheat will be on each square?")

grains = 1

total = 1

for n = 2 to 6

grains = grains \* 2

total = total + grains

print(total)

next n

|  |  |  |
| --- | --- | --- |
| **grains** | **Total** | **n** |
| 1 | 1 | 2 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

OUTPUT:

Task 5

Draw a trace table to follow through this algorithm. State the output if the user enters

21, 10, 6, 5, 30, 9

total = 0

for n = 1 to 6

a = input("Input next value: ")

if a < 10

b = a / 10

print(b)

total = total + b

endif

next n

print(total)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **total** | **n** | **a** | **b** | **OUTPUT** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

What does the algorithm do?

Task 5

The following program asks the user to enter three numbers. It should then calculate the largest of the three numbers. There is a logical error in the code.

a = input("First number: )

b == input("Second number: ")

c = input("Third number: ")

if a > b OR a > c

print(a)

elif b > a AND b > c

print(b

else

print(c)

endif

(a) Identify the error and correct it.

(b) There are **three** syntax errors in the code. Identify these and fix them.