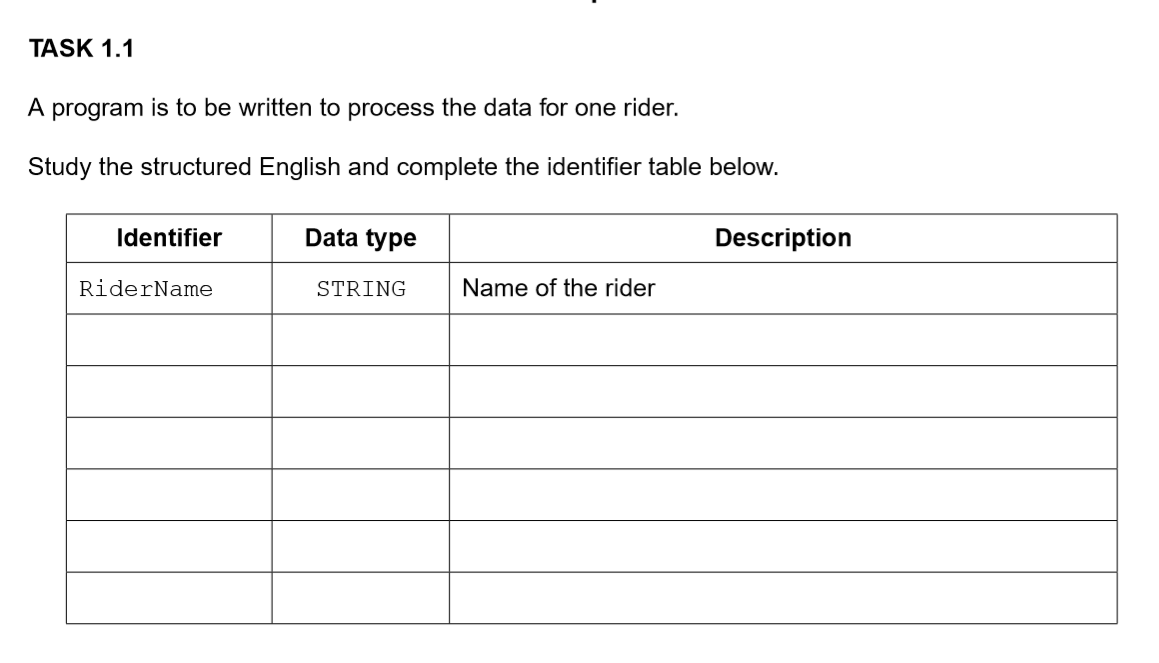
1



RaceTimeMin INTEGER Race time of the rider in minutes

RaceTimeSec INTEGER Race time of the rider in seconds

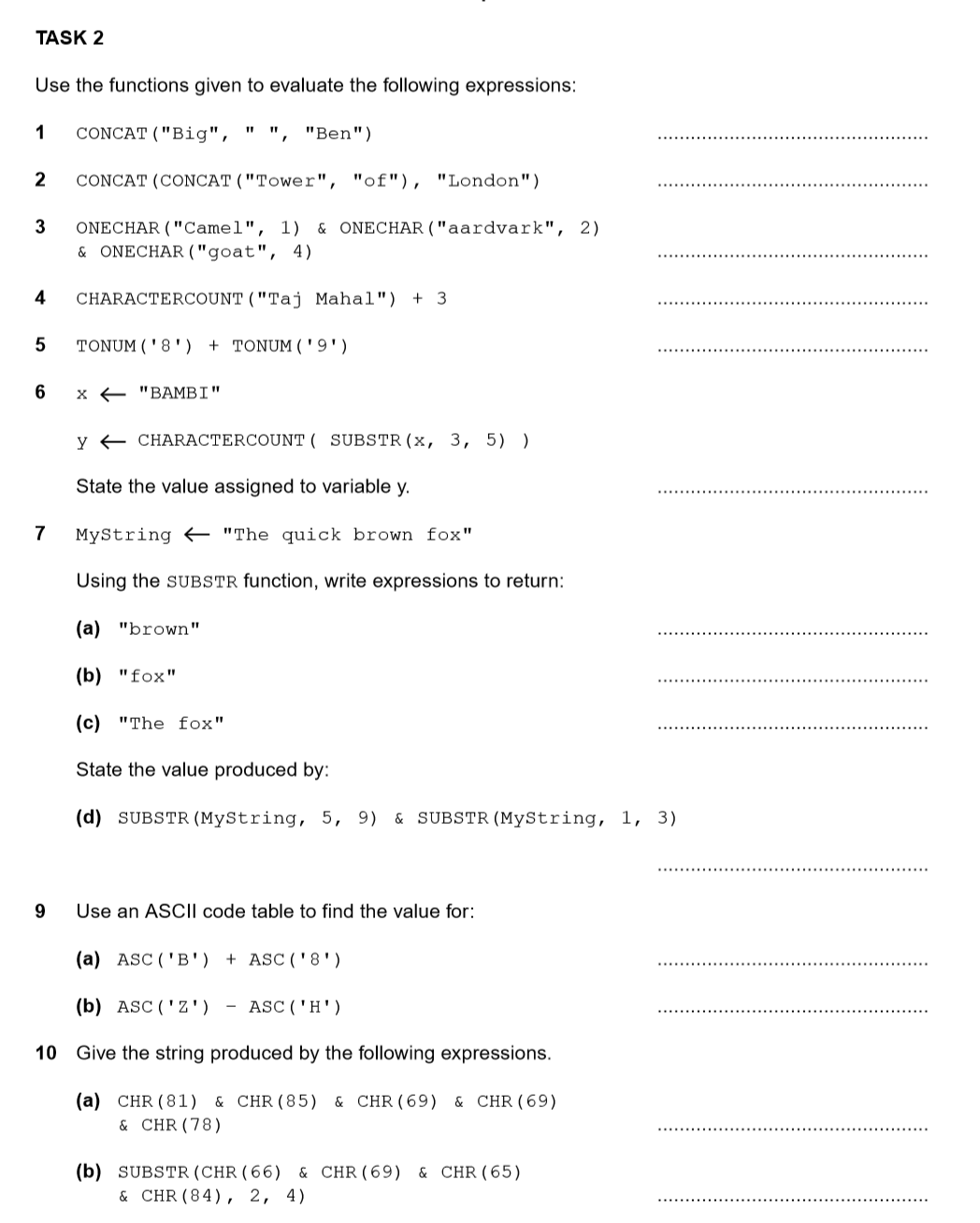
RiderPosition INTEGER Position of the rider

TimeBonus INTEGER Time bonus

AdjustedRaceTime INTEGER Adjusted race time in seconds

**TASK 1.2**

08 LOAD time bonus



122

18

“QUEEN”

“EAT”

SUBSTR(MyString, 11, 15)

SUBSTR(MyString, 17, 19)

SUBSTR(MyString, 1, 3)&SUBSTR(MyString, 16, 19)

“quickThe”

“Big Ben”

“TowerofLondon”

“Cat”

11

17

3

2

**Task 1.1**

“Comp”

“put”

“put”

**Task 1.2**

Let the string as MyString

print(“First name: ”+ MyString[:4]+“, Second name: ”+ MyString[-6:])

print(“First name: Abdul, Second name: ”+ MyString[-6:])

print(“First name: ”+ MyString[:5]+“, Second name: ”+ MyString[-5:])

print(“First name: Abdul, Second name: ”+ MyString[-6:])

print(“First name: Abdul, Second name: ”+ MyString[-6:])

print(MyString)

print(“First name: ”+ MyString[:5]+“, Second name: ”+ lower(MyString[-5:]))

**Task 2.1**

DECLARE UserID : STRING

DECLARE Temp : INTEGER

DECLARE i : INTEGER

DECLARE Flag : BOOLEAN

OUTPUT “Enter an user ID: ”

INPUT UserID

Flag = TRUE

FOR i 🡨 1 TO 6

Temp = ASC(UserID[i])

IF i == 1 AND 65<=Temp<=90

THEN

CONTINUE

ENDIF

IF 2<=i<=3 AND 97<=Temp<=122

THEN

CONTINUE

ENDIF

IF i >=4 AND 48<=Temp<=57

THEN

CONTINUE

ELSE

Flag = FALSE

ENDIF

ENDFOR

IF Flag == TRUE

THEN

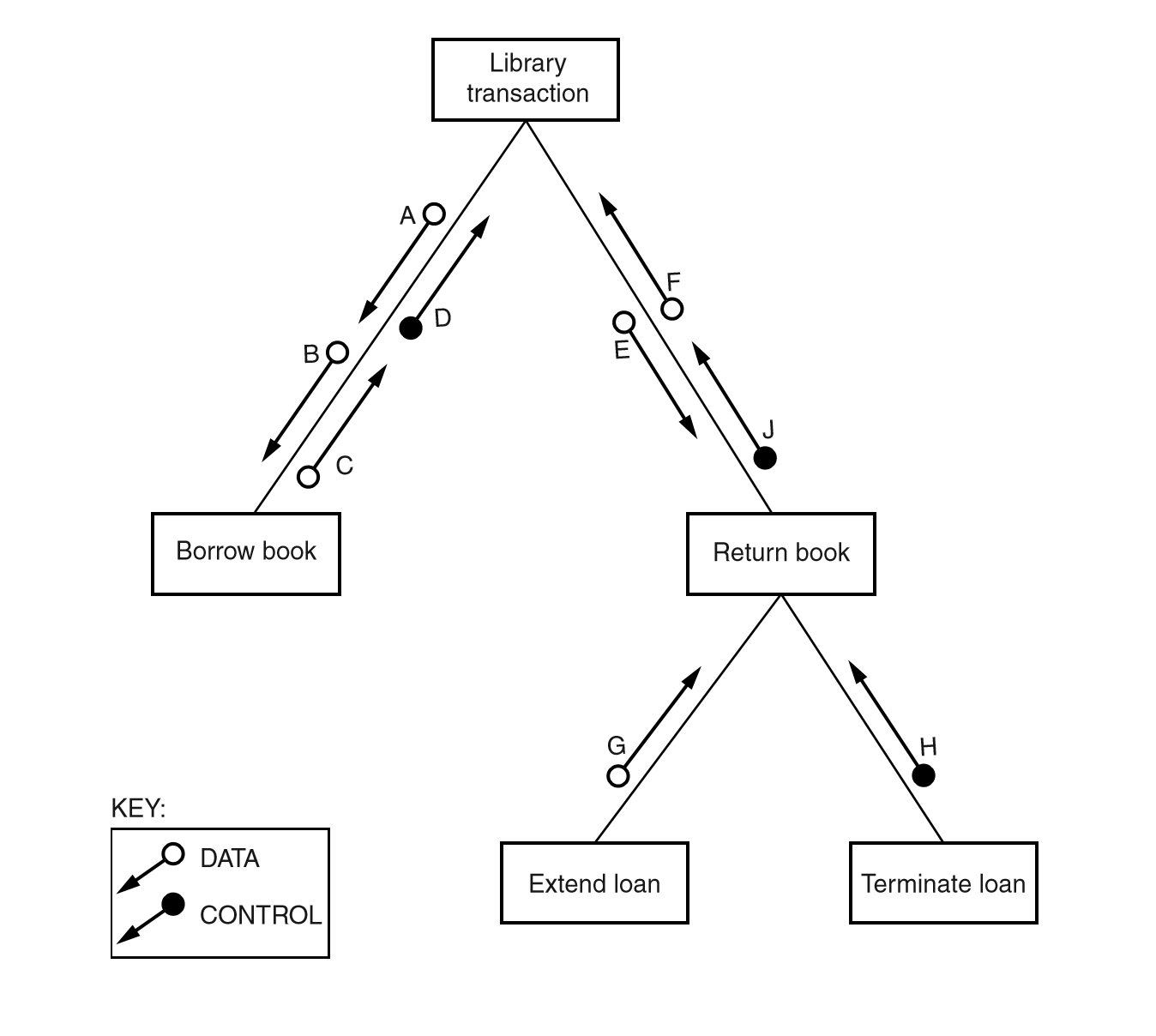
OUTPUT “Wrong Format”

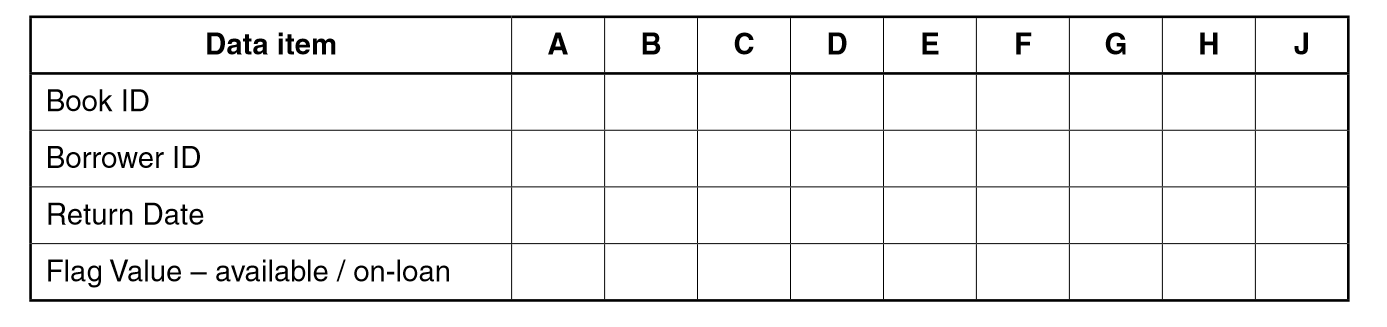
ELSE

OUTPUT “Correct Format”

ENDIF

**Task 4.1**

****

**Task 4.2**

√

√

√

√

√

√

√

√

3

**Task 1.1**

Producing

Studying

Calculating

**Task 1.2**

OUTPUT

FUNCTION

CALCULATE

**Task 1.3**

FALSE

FALSE

TRUE

TRUE

FALSE

FALSE

FALSE

**Task 2.1**

Count = 0

REPEAT

Count = Count + 1

OUTPUT Count

UNTIL Count = 100

WHILE Count < 100

Count = Count + 1

ENDWHILE

**Task 2.2**

OUTPUT “Enter a start value: ”

INPUT StartValue

OUTPUT “Enter an end value: ”

INPUT

FOR Count 🡨 StartValue TO EndValue

OUTPUT MOD(Count, 3)

ENDFOR

**Task 3.1**

DECLARE List1 ARRAY[1:10] OF STRING

**Task 3.2**

REPEAT List1

ADD all strings as a string, FinalString

OUTPUT FinalString

**Task 3.3**

FinalString = “”

FOR i 🡨 1 TO 10

IF List1[i]<> "(Empty)"

THEN

String = List1[i]

FinalString = FinalString + String

ENDIF

ENDFOR

OUTPUT FinalString

4

**Task 1.1**

ADD clubs or hobbies

GO college or school

GO home or factory

**Task 1.4**

IF identifier > UpperLimit

THEN

Statement1

IF identifier > LowerLimit

THEN

Statement2

IF identifier == 5

THEN

Statement3

Statement4

ELSE

Statement5

ENDIF

5

**Task 1.1**

OPENFILE “BOOK-FILE” FOR READ

WHILE NOT EOF(“BOOK-FILE”)

READFILE “BOOK-FILE”, Temp

OUTPUT NextLine

APPEND List1, Temp

ENDWHILE

CLOSEFILE

OUTPUT List1

**Task 2.2**

Start

OPENFILE "BOOK-FILE" FOR READ

IsFound ← FALSE

OUTPUT "Enter book"

INPUT ThisBook

READFILE "BOOK-FILE", FileBook

FileBook = ThisBook?

YES NO

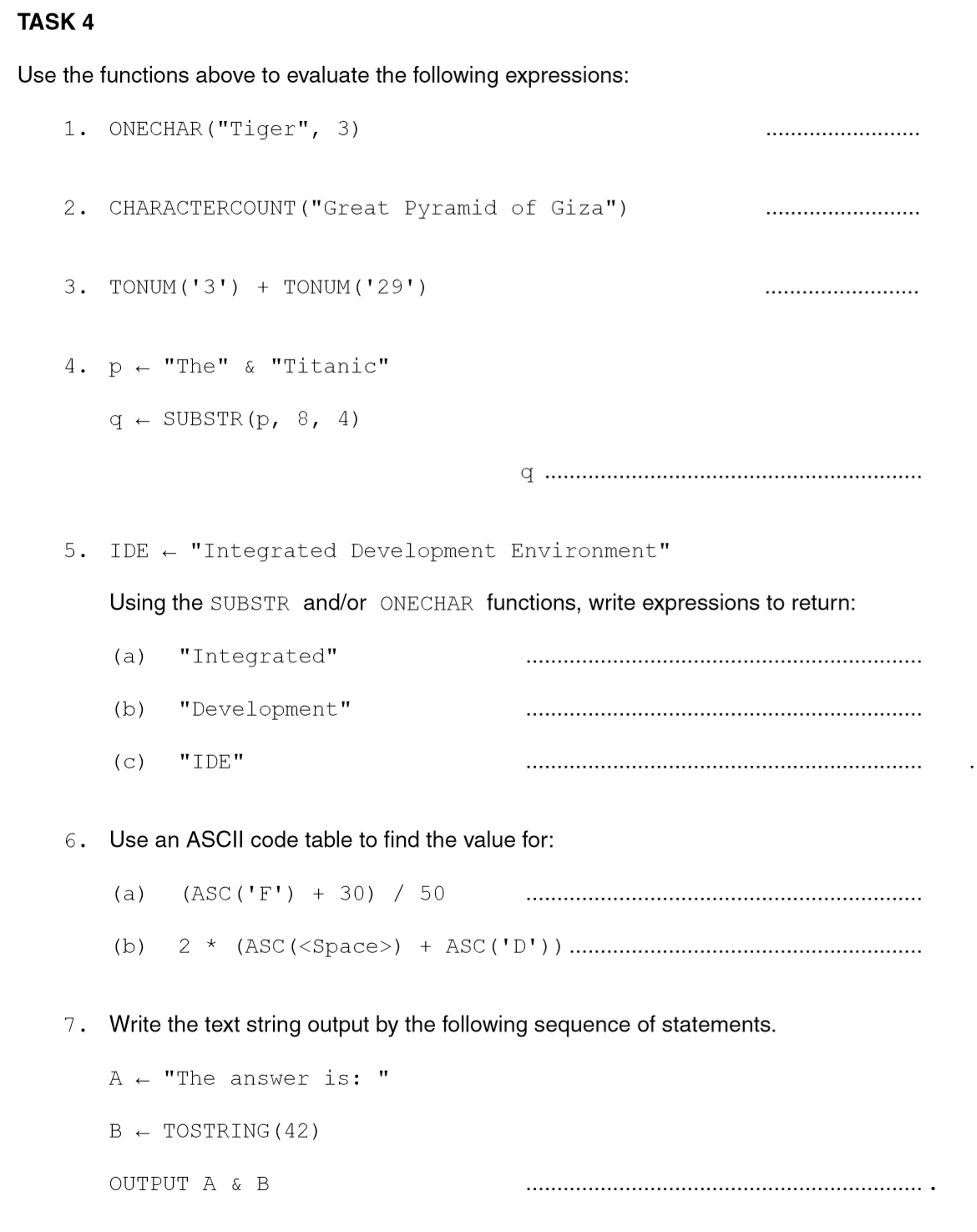
IsFound ← TRUE CONTINUE

OUTPUT "BOOK FOUND"

IsFound = FALSE?

NO YES

END OUTPUT "BOOK NOT FOUND"



2

200

"The answer is: 42"

“g”

21

32

ERROR

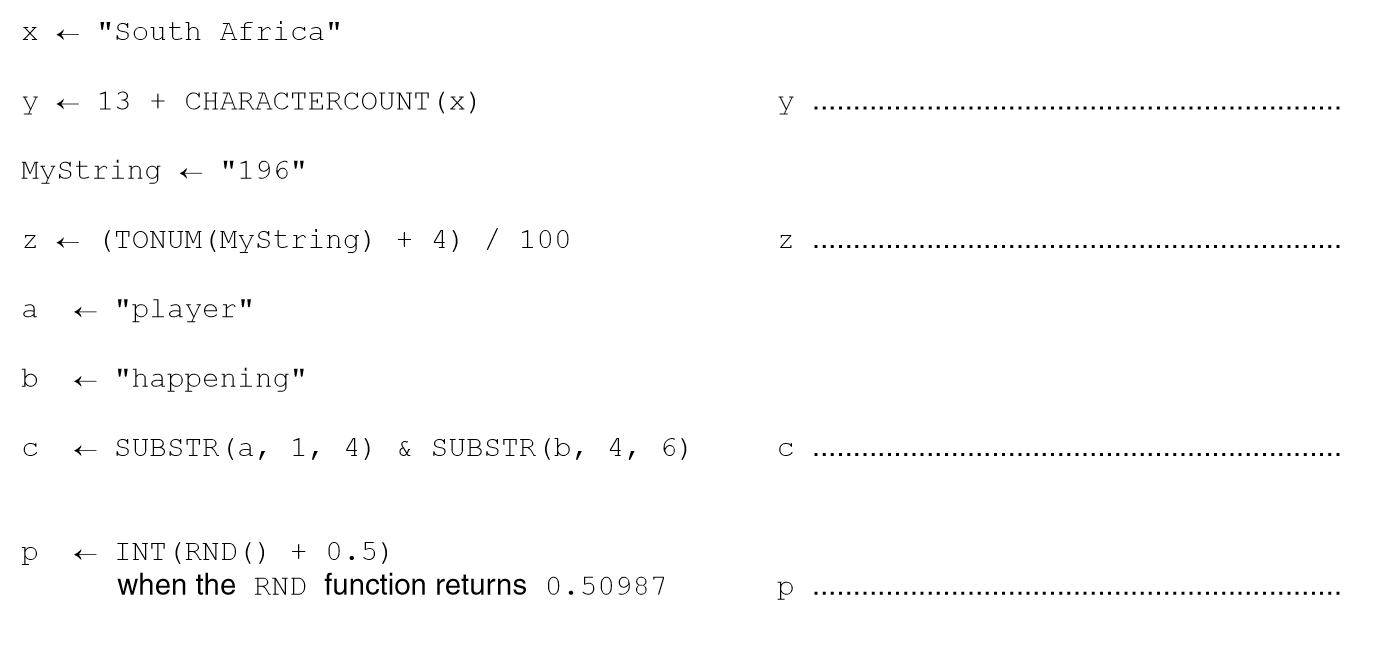
SUBSTR(IDE, 1, 10)

SUBSTR(IDE, 12, 11)

ONECHAR(IDE, 1) & ONECHAR(IDE, 12) & ONECHAR(IDE, 24)

6

**TASK 1**



25

2

“playpening”

1.00987

7

**TASK 1.1**

OUTPUT “Enter the day: ”

INPUT Day

IF Day = “Monday”

THEN

OUTPUT “Red socks”

IF Day = “Tuesday”

THEN

OUTPUT “Blue socks”

IF Day = “Wednesday”

THEN

OUTPUT “Yellow socks”

ELSE

OUTPUT “Sandals”

ENDIF

**TASK 1.2**

OUTPUT “Enter the day: ”

CASE OF INPUT Day

“Monday” : OUTPUT “Red socks”

“Tuesday” : OUTPUT “Blue socks”

“Wednesday” : OUTPUT “Yellow socks”

OTHERWISE OUTPUT “Sandals”

ENDCASE

**TASK 1.3**

"Cold"

"Cold"

"Just right"

"Just right"

"Hot" "Just right"

**TASK 1.4**

ELIF