GCSE COMPUTER SCIENCE

(8525)

Additional Practice questions

Additional practice programming questions including example answers, handwritten student answers and examiner commentary.

Version 1.1

ADDITIONAL PRACTICE QUESTIONS

This resource gives teachers and pupils guidance on the level of detail, syntactic correctness and programming accuracy required in the examinations.

The questions are divided into low, medium and high tariff sections.

Whilst there are separate examination papers for each of the three languages (C#, VB.NET and Python) we have merged the languages and example solutions into a single document so it is possible to see the way solutions in each language will be equally treated when marked.

As detailed in the mark scheme, the case of text will be ignored and indentation will only be taken account of in so far as the logic flow must be clear. Similarly, if punctuation is missing (eg semicolons, colons etc) marks can be awarded as long as the logic is clear.

Question 3 is deliberately contrived to allow us to show that any correct solution to a question will gain marks whether it directly maps to the examples in the mark scheme or not. As long as the solution does what the question requires the marks will be awarded.

Low tariff questions

Q1

Write a Python program that will tell you how old you will be on your next birthday.

Your program should:

- prompt you to enter your age
- add 1 to the entered age
- output your age on your next birthday.

You **should** use meaningful variable name(s), correct syntax and indentation in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

[5 marks]

Mark scheme

Q	Marking guidance	Total marks
Q1	**1 mark for AO3 (design) and 3 marks for AO3 (program)** **Program design** **Mark A** for using meaningful variable names throughout (even if logic is incorrect); **Program logic** **Mark B** for getting user input for the age in an appropriate place; **Mark C** for correctly adding 1 to the inputted age; **Mark D** for outputting the correct final age; **I.** Case of program code **Maximum 3 marks** if any errors in code.	4

Python example 1 (fully correct)

Mark A awarded**.**

```
age = int(input())                         (B)
age = age + 1                              (C)
print(age)                                 (D)
```

C# example (fully correct)

Mark A awarded**.**

```
int age;
age = int.Parse(Console.ReadLine());       (B)
age = age + 1;                             (C)
Console.WriteLine(age);                    (D)
```

I. indentation in C#

VB example (fully correct)

Mark A awarded**.**

```
Dim age As Integer
age = Console.ReadLine()                   (B)
age = age + 1                              (C)
Console.WriteLine(age)                     (D)
```

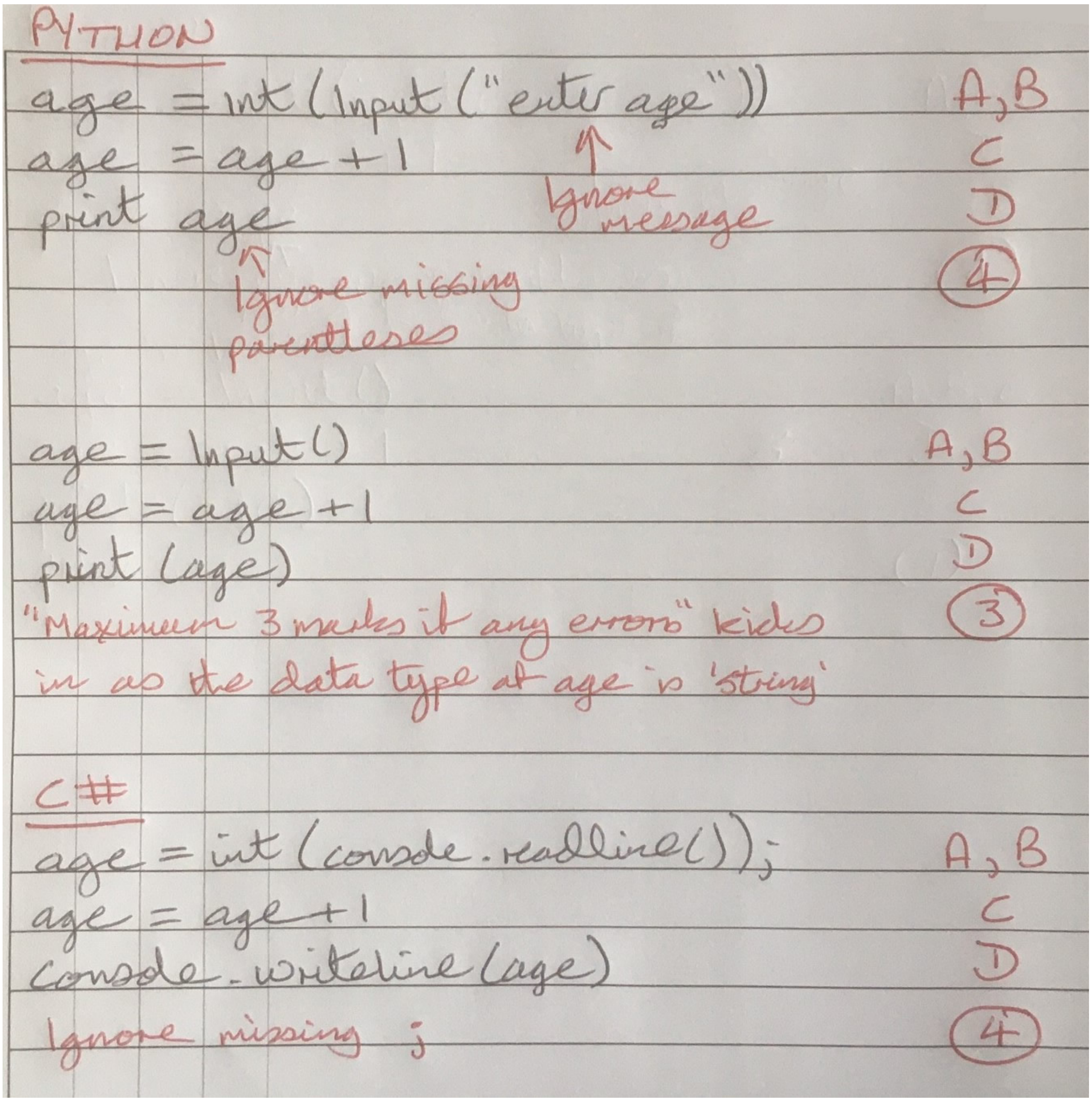
I. indentation in VB.NET

Python example 2 (partially correct – 3 marks)

Mark A awarded**.**

```
age = input()               (B)
age = age + 1               (C)
print age                   (D – still awarded
                            even though parentheses missing
                            in print command as logic still clear)
```

'Maximum 3 marks if any errors in code' is enforced because int conversion is missing for the inputted value. Python defaults to inputting a string.



PYTHON

```
age = int(Input("enter age"))      A, B
age = age + 1                      C
print age                          D
                                   4
```

Ignore message

Ignore missing parentheses

```
age = Input()                      A, B
age = age + 1                      C
print (age)                        D
                                   3
```

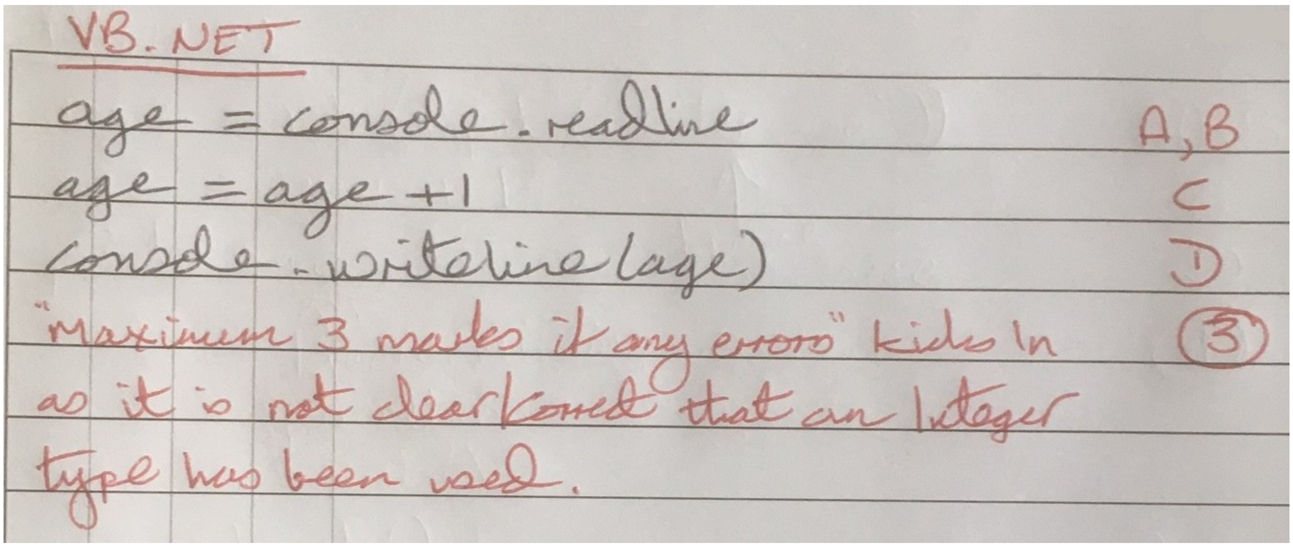
"Maximum 3 marks if any errors" kicks in as the data type of age is 'string'

C#

```
age = int (console.readline());    A, B
age = age + 1                      C
console.writeline(age)             D
                                   4
```

Ignore missing ;

Note: whilst `int(console.readline())` would not work in C#, as it should have been written as `int.Parse(Console.Readline())`, the pupil's intention is clear and the omission is classed as a minor syntax error that would not be penalised.



VB.NET

```
age = console.readline                A,B
age = age +1                          C
console.writeline(age)                D
```

"Maximum 3 marks if any errors" kicks in as it is not clear/correct that an Integer type has been used. $\boxed{3}$

Note: Missing () in VB on Readline and case are ignored.

Q2

Write a Python program that will calculate the volume of a rectangular swimming pool with a depth of two metres. The formula for calculating the volume is:

volume = length x width x depth

Your program should:

- prompt the user to enter the length in metres (the value should be a whole number)
- prompt the user to enter the width in metres (the value should be a whole number)
- calculate the correct volume
- output the volume.

You **should** use meaningful variable name(s), correct syntax and indentation in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

[5 marks]

Mark scheme

Q	Marking guidance	Total marks
Q2	**1 mark for AO3 (design) and 3 marks for AO3 (program)** **Program design** **Mark A** for using meaningful variable names throughout (even if logic is incorrect); **Program logic** **Mark B** for getting user input for both length and width in an appropriate place; **Mark C** for correctly calculating the volume; **Mark D** for outputting the final volume; **I.** Case of program code **Maximum 3 marks** if any errors in code.	4

Python example 1 (fully correct)

Mark A awarded**.**

`length = int(input())`	(**Part B**)
`width = int(input())`	(**Part B**)
`volume = length * width * 2`	(**C**)
`print(volume)`	(**D**)

C# example (fully correct)

Mark A awarded**.**

`int length;`	
`int width;`	
`length = int.Parse(Console.ReadLine());`	(**Part B**)
`width = int.Parse(Console.ReadLine());`	(**Part B**)
`volume = length * width * 2`	(**C – missing ; not penalised**)
`Console.WriteLine(volume);`	(**D**)

I. indentation in C#

VB example (fully correct)

Mark A awarded**.**

`Dim length As Integer`	
`Dim width As Integer`	
`length = Console.ReadLine()`	**(Part of B**)
`width = Console.ReadLine()`	**(Part of B)**
`volume = length * width * 2`	(**C**)
`Console.WriteLine(volume)`	(**D**)

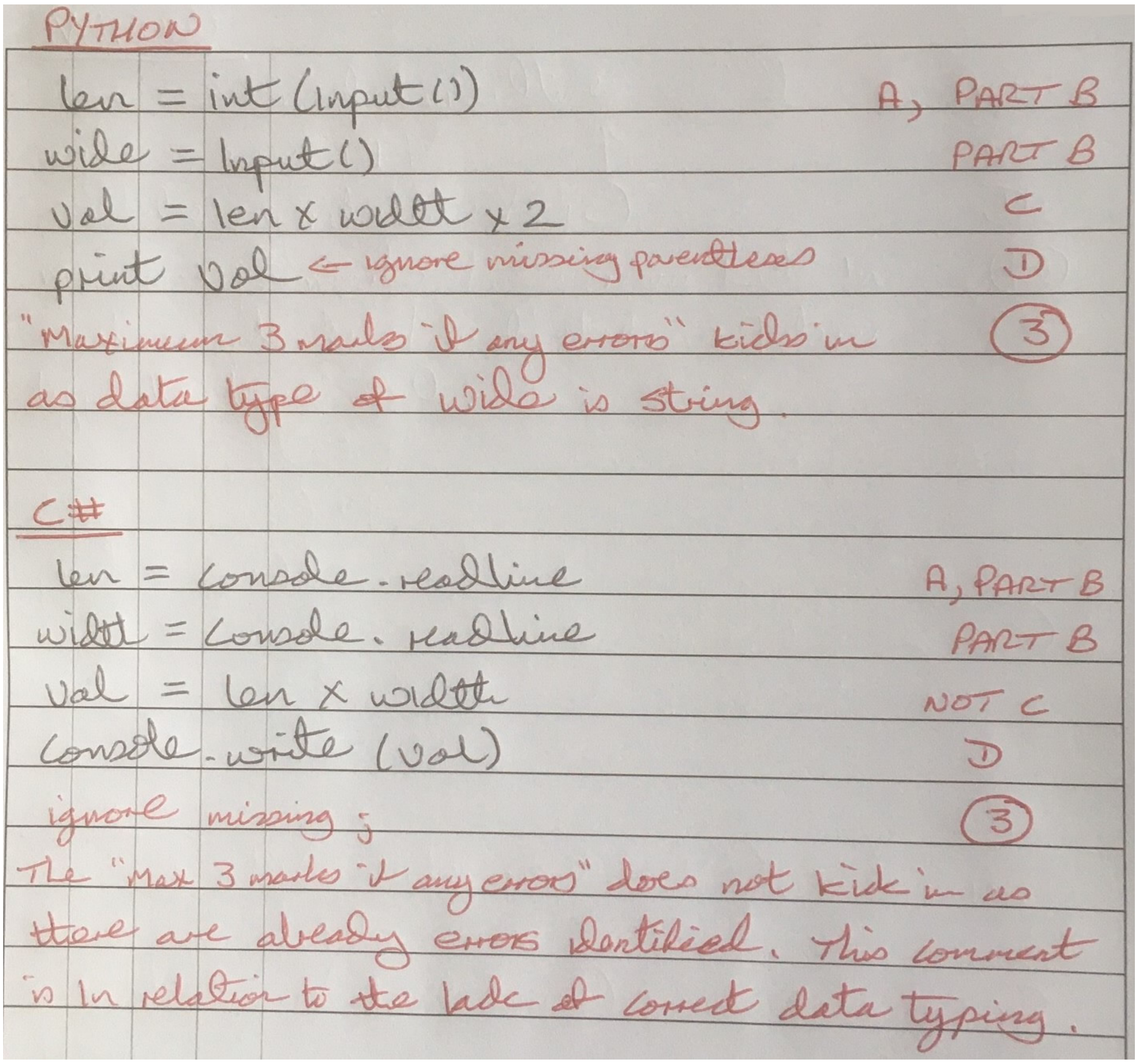
I. indentation in VB.NET

Python example 2 (partially correct – 3 marks)

Mark A awarded**.**

`length = int(input())`	(**Part B**)
`width = input()`	(**Part B**)
`volume = length * width`	(**Not C**)
`print volume`	(**D – still awarded even though parentheses missing in print command as logic still clear**)

If this code had received all mark points then the **'Maximum 3 marks** if any errors in code' would have been enforced because int conversion is missing for the second inputted value. However, as the code already contains an error that resulted in mark C not being awarded this additional issue can be ignored.



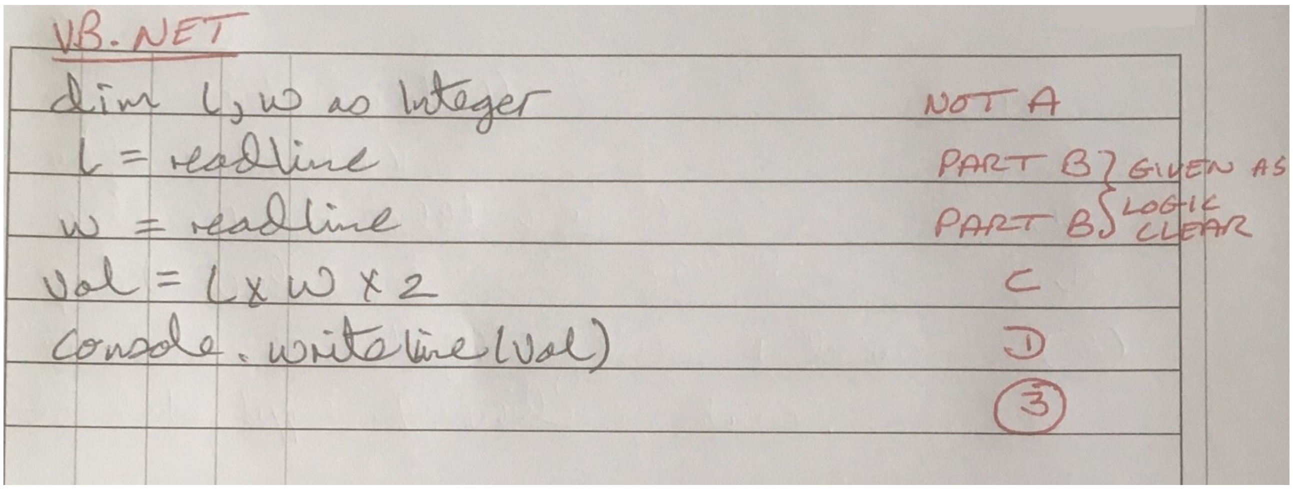
PYTHON

len = int(input())	A, PART B
wide = input()	PART B
val = len x width x 2	C
print val ← ignore missing parentheses	D
"Maximum 3 marks if any errors" kicks in as data type of wide is string.	③

C#

len = console.readline	A, PART B
width = Console.readline	PART B
val = len x width	NOT C
console.write (val)	D
ignore missing ;	③

The "Max 3 marks if any errors" does not kick in as there are already errors identified. This comment is in relation to the lack of correct data typing.



VB.NET

dim l, w as integer	NOT A
l = readline	PART B } GIVEN AS LOGIC CLEAR
w = readline	PART B
val = l x w x 2	C
console.writeline(val)	D
	③

Mid tariff question

Q3

The OR logic gate outputs a 1 if either of the two inputs are 1, otherwise it will output a 0

For example:

- if the two inputs are 0 and 1 then it will output a 1
- if the two inputs are both 0 then it will output a 0

Write a Python program that will output the result of performing an OR logic gate.

Your program should:

- keep asking the user to enter two values until they enter two values, each of which must be either a 0 or a 1
- calculate the correct output from an OR gate using the two inputs that have been entered
- output the result of the OR gate.

You **should** use meaningful variable name(s), correct syntax and indentation in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

[7 marks]

Mark scheme

Q	Marking guidance	Total marks
Q3	**2 marks for AO3 (design) and 5 marks for AO3 (program)** **Program design** **Mark A** for using meaningful variable names throughout (even if logic is incorrect); **Mark B** for attempting to use indefinite iteration (even if logic is incorrect); **Program logic** **Mark C** for getting user input for both logic gate inputs in an appropriate place; **Mark D** for correctly re-entering one or both of the inputs when required (even if the Boolean condition is incorrect); **Mark E** for a correct Boolean condition to validate one or both of the user inputs; **Mark F** for any method that correctly performs an OR gate operation on the two inputs **Mark G** for outputting the final result of an OR gate operation on the two inputs; **I.** Case of program code **Maximum 6 marks** if any errors in code.	7

Python example 1 (fully correct)

Marks A and **B** awarded**.**

```
valid = False                                   (Part E)
while not valid:
    input1 = int(input())                       (Part C, Part D)
    if input1 == 1 or input1 == 0:
        valid = True                            (Part E)
valid = False
while not valid:
    input2 = int(input())                       (Part C, Part D)
    if input2 == 1 or input2 == 0:
        valid = True
if input1 != input2:                            (F)
    result = 1
elif input1 == 1 and input2 == 1:
    result = 1
else:
    result = 0
print(result)                                   (G)
```

C# example (fully correct)

Mark A and **B** awarded.

```
int result;
bool valid=false;                              (Part E)
int input1 = 2, input2 = 2;
while (valid == false)
{
  input1 = int.Parse(Console.ReadLine());    (Part C, Part D)
  if (input1 == 1 | input1 == 0)
  { valid = true; }                          (Part E)
}
valid = false;
while (valid == false)
{
  input2 = int.Parse(Console.ReadLine());    (Part C, Part D)
  if (input2 == 1 | input2 == 0)
  { valid = true; }
}
result = input1 | input2                     (F)
Console.WriteLine(result);                   (G)
```

I. indentation in C#

VB example (fully correct)

Mark A and **B** awarded.

```
Dim result As Integer
Dim valid As Boolean
Dim input1, input2 As Integer
valid = False                               (Part E)
While valid = False
  input1 = Console.ReadLine()               (Part C, Part D)
  If input1 = 1 Or input1 = 0 Then
    valid = True                            (Part E)
  End If
End While
valid = False
While valid = False
  input2 = Console.ReadLine()               (Part C, Part D)
  If input2 = 1 Or input2 = 0 Then
    valid = True
  End If
End While
result = input1 or input2                   (F)
Console.WriteLine(result)                   (G)
```

I. indentation in VB.NET

Python example 2 (partially correct – 6 marks)

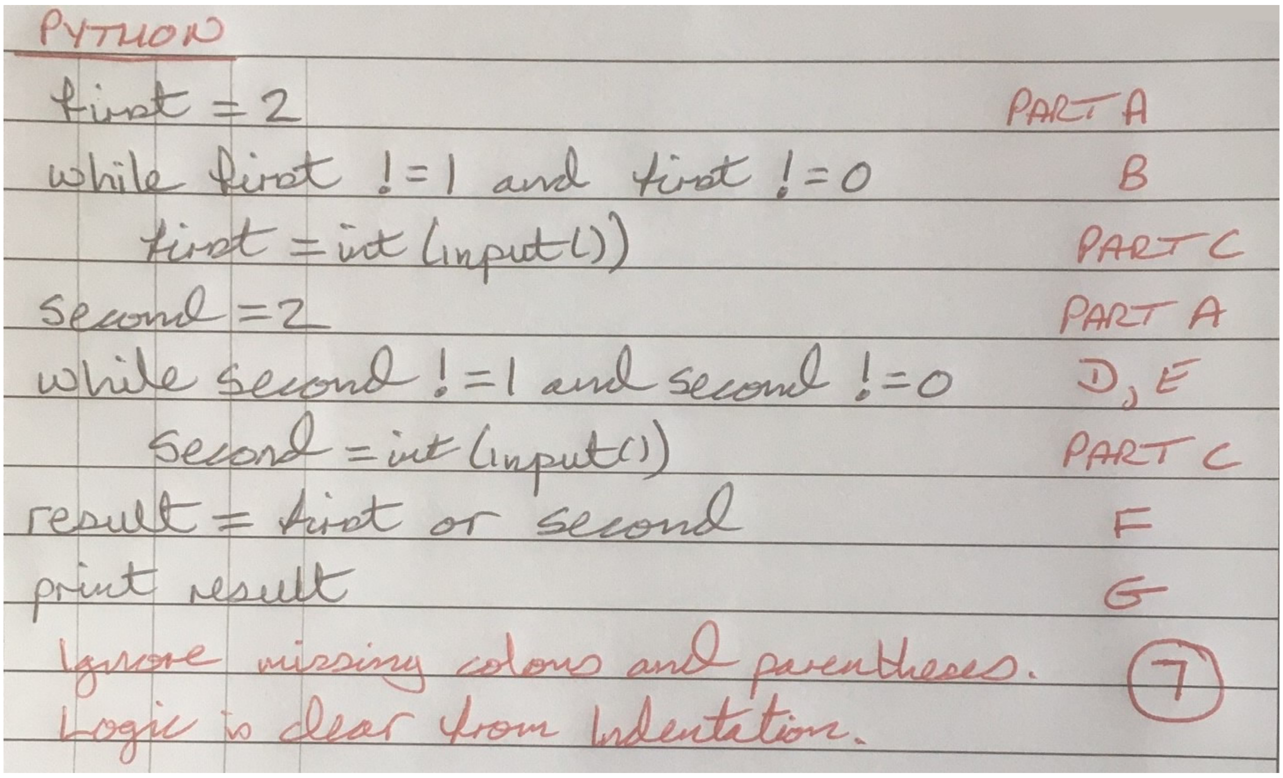
Mark A and **B** awarded.

```
v = False                              (Part E)
while not v:
    input1 = input                     (Part C – still awarded even though
                                       parentheses missing in input
                                       command as logic still clear,
                                       Part D)
    if input1 == 1 or input1 == 0:
        v = True                       (Part E)
v = False
while not v:
    input2 = input()                   (Part C, Part D)
    if input2 == 1 or input2 == 0:
        v = True

if input1 <> input2:                   (F – not awarded as logic is not
                                       correct in elif part.  The use of
                                       <> would not have resulted in the
                                       loss of the mark if all other logic
                                       had been correct as the use of <>
                                       in place of != does not affect the
                                       overall clarity of the logic)
    result = 1
elif input1 == 1 and input2 == 0:
    result = 1
else:
    result = 0
print result                           (G – still awarded even though
                                       parentheses missing in print
                                       command as logic still clear)
```

If this code had received all mark points then the **'Maximum 6 marks** if any errors in code' would have been enforced because int conversion is missing for the inputted values. However, as the code already contains an error that resulted in mark F not being awarded this additional issue can be ignored.

Student answers with examiner commentary

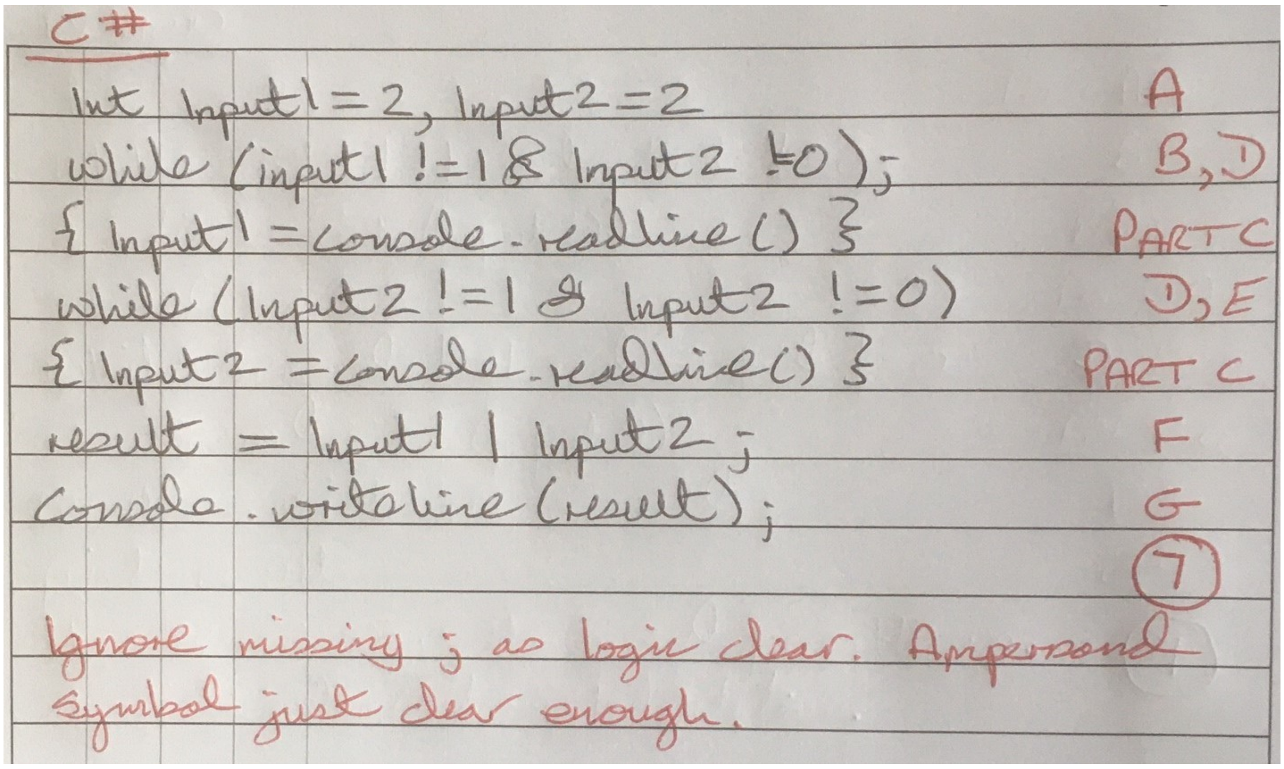
Python



```
PYTHON
first = 2                                      PART A
while first != 1 and first != 0                B
    first = int (input())                      PART C
second = 2                                     PART A
while second != 1 and second != 0              D, E
    second = int (input())                     PART C
result = first or second                       F
print result                                   G
```

Ignore missing colons and parentheses. ⑦
Logic is clear from Indentation.

C#

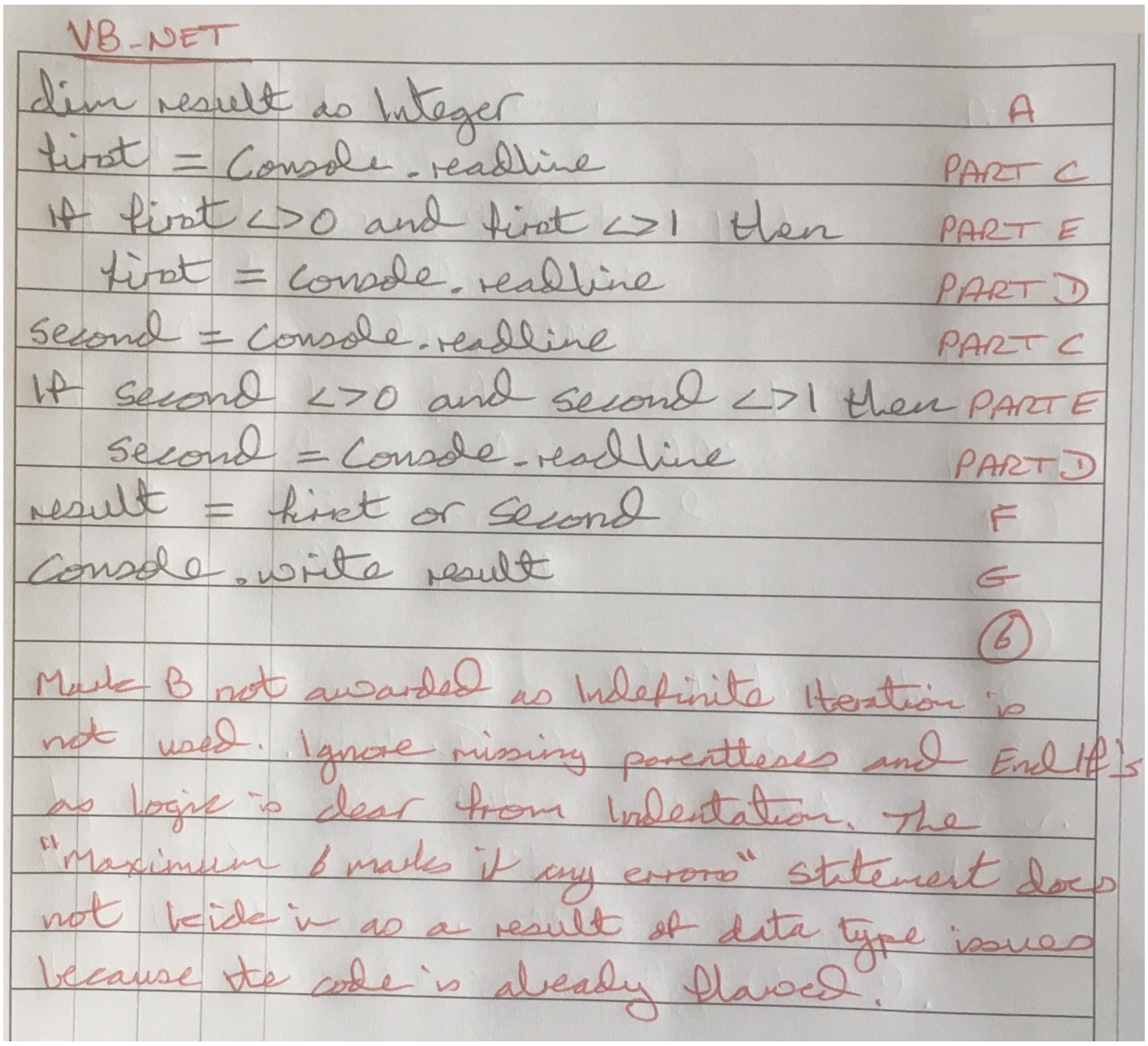


```
C#
 Int Input1 = 2, Input2 = 2                    A
 while (input1 != 1 & Input2 != 0);            B, D
 { Input1 = console.readline() }               PART C
 while (Input2 != 1 & Input2 != 0)             D, E
 { Input2 = console.readline() }               PART C
 result = Input1 | Input2 ;                    F
 Console.writeline (result);                   G
```

⑦

Ignore missing ; as logic clear. Ampersand symbol just clear enough.

VB.NET



VB.NET

```
dim result as Integer                                  A
first = Console.readline                               PART C
If first <>0 and first <>1 then                        PART E
    first = console.readline                           PART D
Second = console.readline                              PART C
If second <>0 and second <>1 then                      PART E
    second = Console.readline                          PART D
result = first or Second                               F
Console.write result                                   G
                                                       6
```

Mark B not awarded as Indefinite Iteration is not used. Ignore missing parentheses and End If's as logic is clear from Indentation. The "Maximum 6 marks if any errors" statement does not kick in as a result of data type issues because the code is already flawed.

High tariff question

Q4

Write a Python program that plays the following number guessing game.

Your program should:

- randomly generate a 2 digit numeric code (ie numbers between 10 and 99)
- allow the user 10 turns to guess the code as follows:
    - prompt the user to enter a 2 digit number (validation is not required)
    - calculate the number of correct digits in the correct place
    - output a suitable message followed by the number of correct digits in the correct place
- output a suitable message if the user has guessed the 2 digit code correctly within 10 turns
- output a suitable message along with the correct code if the user has had 10 turns and failed to guess the code correctly.

To generate a random number between two values you can use the Python command

`random.randrange(x, y)`

This will generate a random integer between x and y - 1 inclusive. For example, the command `random.randrange(2,8)` will generate a random number between 2 and 7.

To generate a random number between two values you can use the C# command

`rnd.Next(x, y);`

This will generate a random integer between x and y - 1 inclusive. For example, the command `rnd.Next(2, 8);` will generate a random number between 2 and 7.

To generate a random number between two values you can use the VB.NET command

`rnd.Next(x, y);`

This will generate a random integer between x and y - 1 inclusive. For example, the command `rnd.Next(2, 8);` will generate a random number between 2 and 7.

You **should** use meaningful variable name(s), correct syntax and indentation in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

The first line of the program has been completed for you.

[10 marks]

```
(Python) import random
(C#) Random rnd = new Random();
(VB) Dim rnd = New Random()
```

Mark scheme

Q	Marking guidance	Total marks
Q4	**2 marks for AO3 (design) and 8 marks for AO3 (program)** **Program Design** **Mark A** for using meaningful variable names throughout (even if logic is incorrect); **Mark B** for attempting to use indefinite iteration (even if logic is incorrect); **Program Logic** **Mark C** for randomly generating a two digit numeric code in an appropriate place (each digit could be generated separately); **Mark D** for allowing multiple turns to be made (even if the Boolean condition is incorrect); **Mark E** for a correct Boolean condition to stop the game once 10 turns have been made or the game has been won; **Mark F** for getting user input for the two digit guess in an appropriate place (each digit could be entered separately); **Mark G** for calculating the number of correct digits in the correct place; **Mark H** for outputting a suitable message followed by the number calculated for **Mark G**; **Mark I** for outputting a suitable message, in an appropriate place, if the game has been won; **Mark J** for outputting a suitable message followed by the correct code, in an appropriate place, if the game has not been won; **I.** Case of program code **Maximum 9 marks** if any errors in code.	10

Python example 1 (fully correct)

Mark A and **B** awarded.

```
codedigit1 = random.randrange(0, 10)            (Part C)
codedigit2 = random.randrange(0, 10)            (Part C)

gameover = False                                (Part E)
noguesses = 0
while not gameover:                             (D, Part E)
    numbercorrect = 0

    userguess = int(input())                    (F)
    noguesses = noguesses + 1
    firstdigit = userguess // 10
    seconddigit = userguess % 10

    if codedigit1 == firstdigit:                (Part G)
        numbercorrect = numbercorrect + 1
    if codedigit2 == seconddigit:               (Part G)
        numbercorrect = numbercorrect + 1

    print("Digits in the correct place: ",numbercorrect)
                                                (H)

    if noguesses == 10 or numbercorrect == 2:
        gameover = True                         (Part E)

if numbercorrect == 2:
    print("Well done you got the code correct")  (I)
else:
    print("Sorry, you failed.  The correct code was:
",codedigit1, codedigit2)                       (J)
```

C# example (fully correct)

Marks A and B awarded.

```
int codedigit1 = rnd.Next(0, 10);                (Part C)
int codedigit2 = rnd.Next(0, 10);                (Part C)

bool gameover = false;                           (Part E)
int noguesses = 0;
int userguess;
int numbercorrect = 0;

while (!gameover)                                (D, Part E)
{
  numbercorrect = 0;

  userguess = int.Parse(Console.ReadLine());     (F)
  noguesses += 1;
  int firstdigit = userguess / 10;
  int seconddigit = userguess % 10;

  if (codedigit1 == firstdigit)                  (Part G)
  { numbercorrect += 1; }
  if (codedigit2 == seconddigit)                 (Part G)
  { numbercorrect += 1; }

  Console.WriteLine("Digits in the correct place: " +
numbercorrect);                                  (H)

  if (noguesses == 10 | numbercorrect == 2)
  { gameover = true; }                           (Part E)
}
if (numbercorrect == 2)
{ Console.WriteLine("Well done"); }              (I)
else
{ Console.WriteLine("Sorry, you failed.  The correct code
was: " + codedigit1 + codedigit2); }             (J)
```

I. indentation in C#

VB example (fully correct)

Marks A and **B** awarded**.**

```
Dim codedigit1, codedigit2 As Integer
codedigit1 = rnd.Next(0, 10)                            (Part C)
codedigit2 = rnd.Next(0, 10)                            (Part C)

Dim gameover As Boolean = False                         (Part E)
Dim noguesses As Integer = 0
Dim userguess, firstdigit, seconddigit As Integer
Dim numbercorrect As Integer = 0


While Not gameover                                      (D, Part E)
  numbercorrect = 0
  userguess = Console.ReadLine()                        (F)
  noguesses += 1
  firstdigit = userguess \ 10
  seconddigit = userguess Mod 10

  If codedigit1 = firstdigit Then                       (Part G)
    numbercorrect += 1
  End If
  If codedigit2 = seconddigit Then                      (Part G)
    numbercorrect += 1
  End If

  Console.WriteLine("Digits in the correct place: " +
Str(numbercorrect))                                     (H)

  If noguesses = 10 Or numbercorrect = 2 Then
    gameover = True                                     (Part E)
  End If
End While

If numbercorrect = 2 Then
  Console.WriteLine("Well done you got the code correct")
                                                        (I)
Else
  Console.WriteLine("Sorry, you failed.  The correct code
was: " + Str(codedigit1) + Str(codedigit2))
                                                        (J)

End If
```

I. indentation in VB.NET

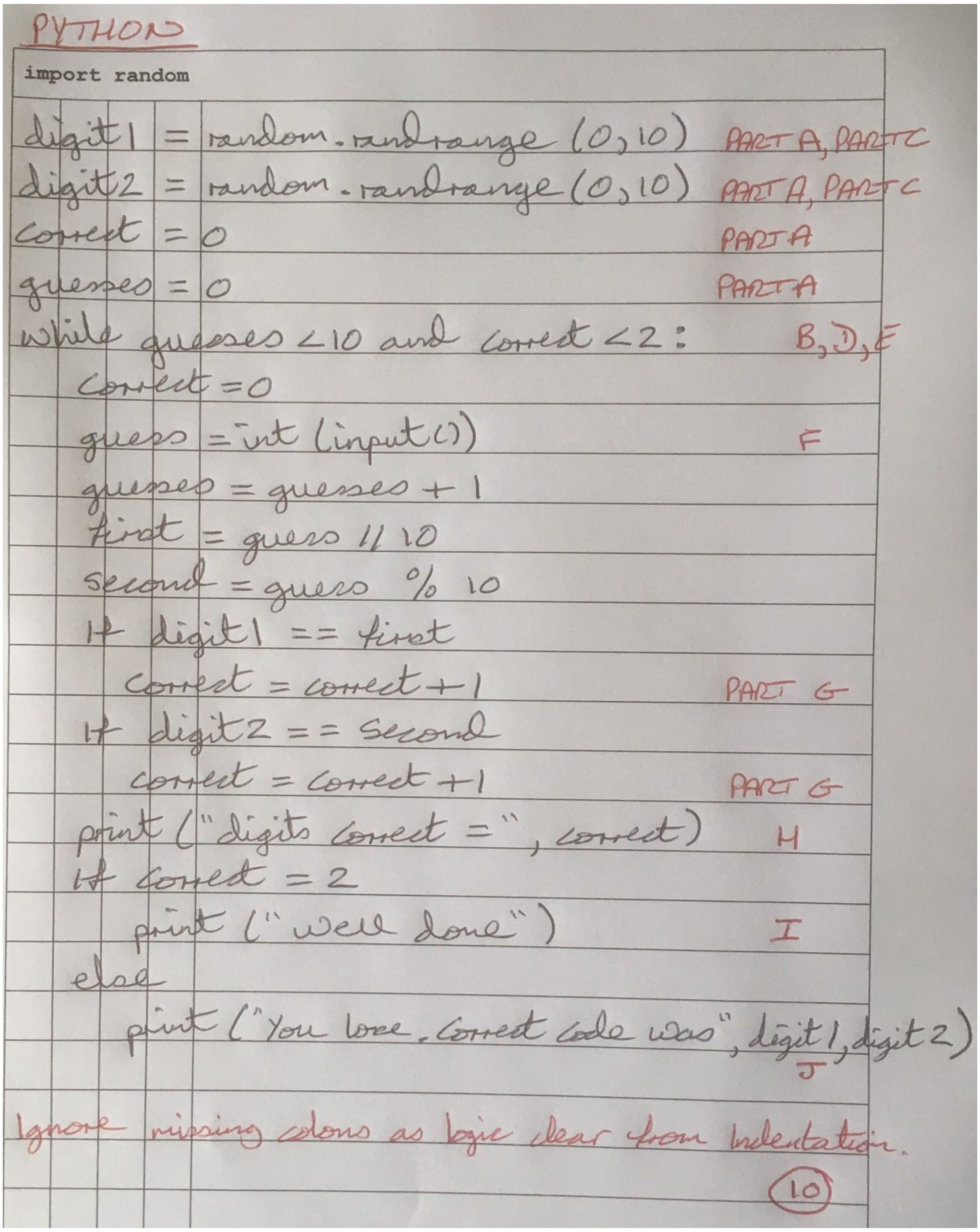
Python example 2 (partially correct – 6 marks)

Mark A awarded**. Mark B** NOT awarded.

```
codedigit1 = random.randrange(0,10)         (Part C)
codedigit2 = random.randrange(0,10)         (Part C)

gameover = False                            (Part E)
noguesses = 0
if not gameover:                            (Not D, Part E)
    numbercorrect = 0

    userguess = input()                     (F)
    noguesses = noguesses + 1
    firstdigit = userguess // 10
    seconddigit = userguess % 10

    if codedigit1 == firstdigit or codedigit2 == seconddigit
        numbercorrect = numbercorrect + 1   (Part G – but mark
                                            not awarded as logic
                                            incorrect)

    print("Digits in the correct place: ",numbercorrect)
                                            (H)

    if noguesses == 10 or numbercorrect == 2:
        gameover = True                     (Part E)

if numbercorrect == 2:
    print("Well done you got the code correct")
                                            (I)
else:
    print("Sorry, you failed.)              (Not J – as message
                                            incomplete.
                                            Ignore missing ")
```

Student answers with examiner commentary

Python



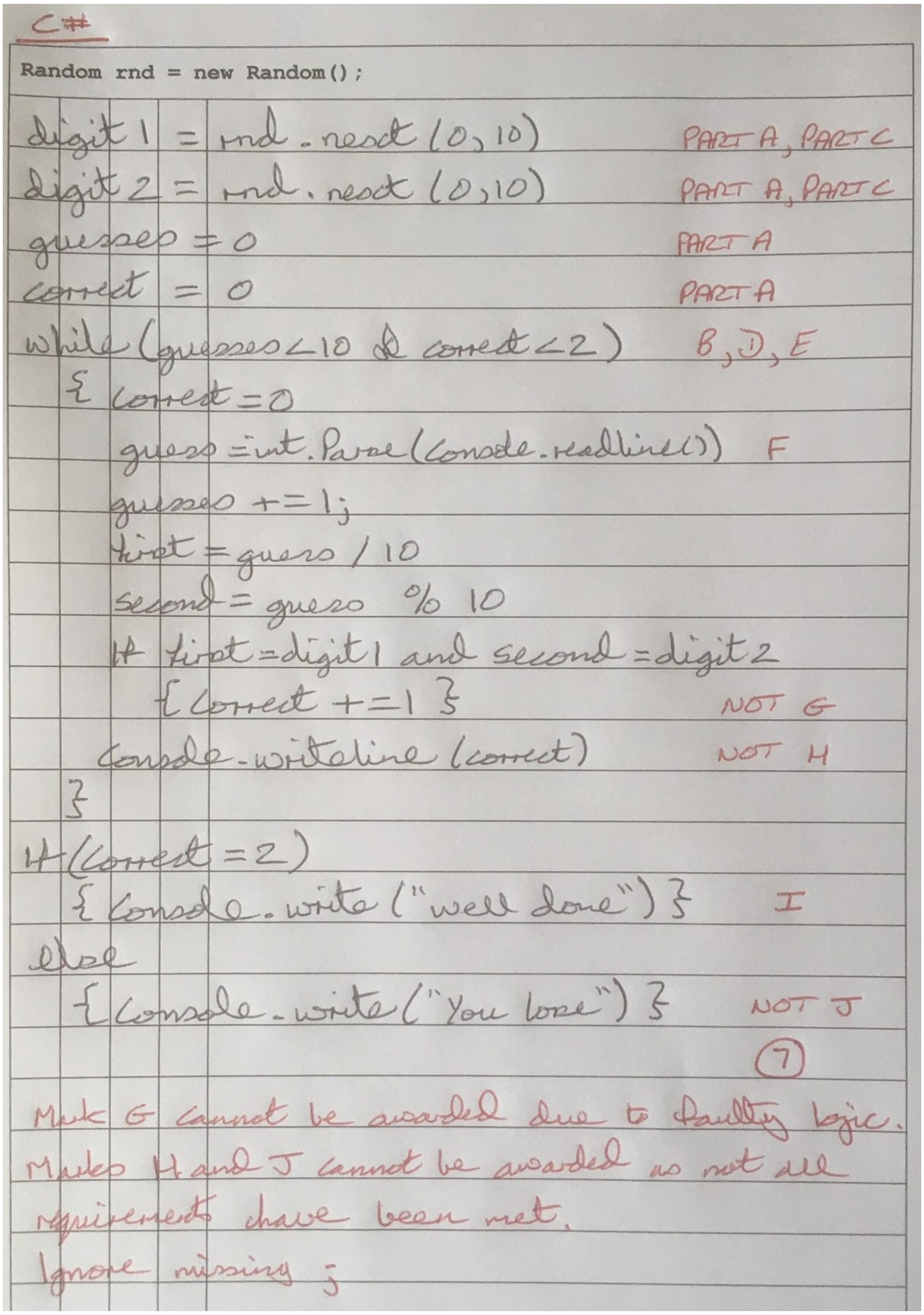
PYTHON

```
import random
digit1 = random.randrange(0,10)                    PART A, PART C
digit2 = random.randrange(0,10)                    PART A, PART C
correct = 0                                        PART A
guesses = 0                                        PART A
while guesses <10 and correct <2:                      B, D, E
    correct =0
    guess = int(input())                               F
    guesses = guesses + 1
    first = guess // 10
    second = guess % 10
    if digit1 == first
        correct = correct +1                       PART G
    if digit2 == second
        correct = correct +1                       PART G
    print("digits correct =", correct)                 H
    if correct = 2
        print("well done")                             I
    else
        print("You lose. Correct code was", digit1, digit2)
                                                 J
```

Ignore missing colons as logic clear from Indentation.

10

C#

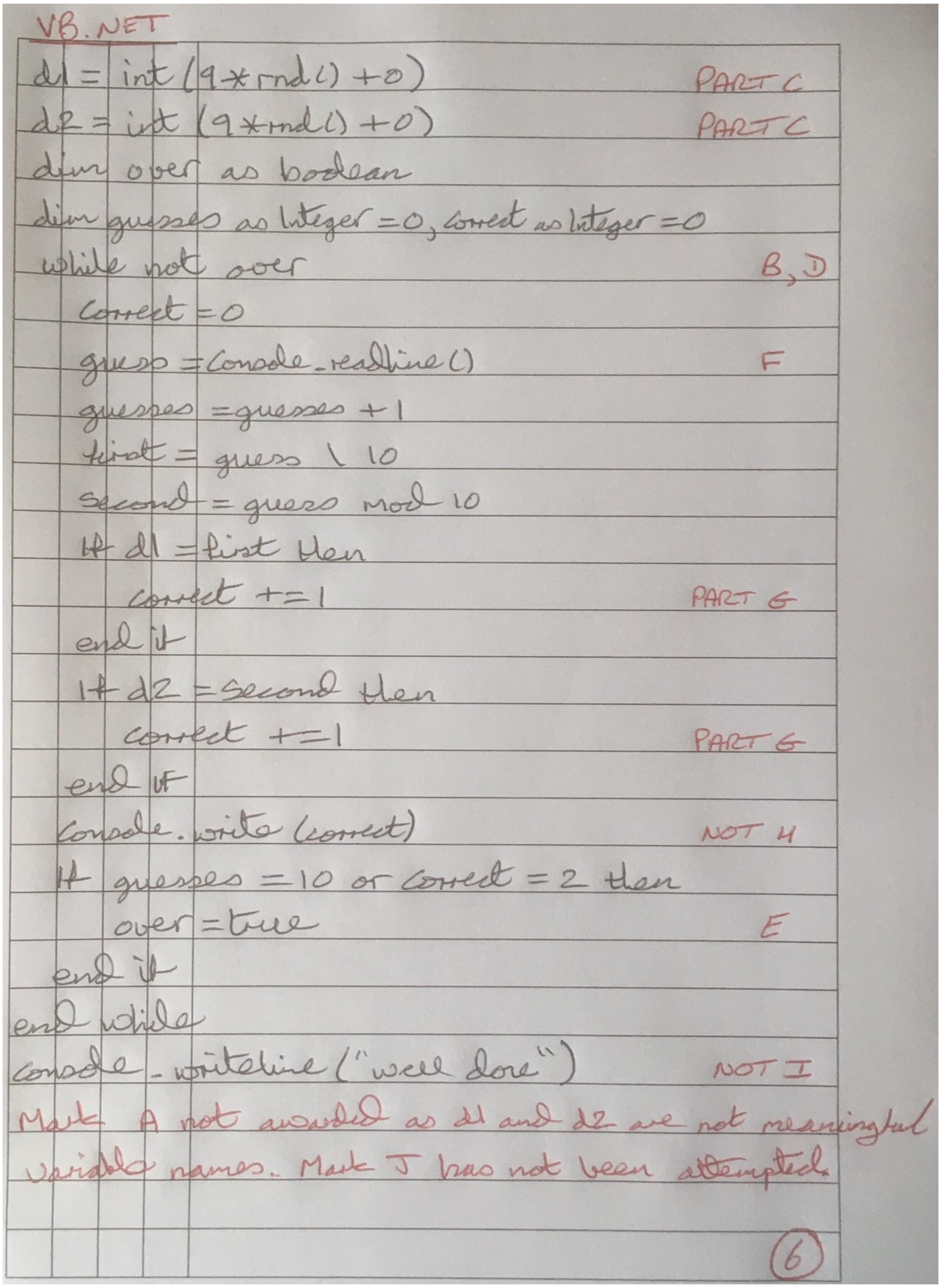


C#

```
Random rnd = new Random();
digit1 = rnd.next(0,10)                          PART A, PART C
digit2 = rnd.next(0,10)                          PART A, PART C
guesses = 0                                      PART A
correct = 0                                      PART A
while (guesses<10 & correct<2)                   B, D, E
  { correct=0
    guess =int.Parse(Console.readline())         F
    guesses +=1;
    first = guess / 10
    second = guess  % 10
    if first=digit1 and second=digit2
      {Correct +=1 }                             NOT G
    Console.writeline (correct)                  NOT H
  }
if (Correct =2)
  { Console.write ("well done") }                I
else
  { Console.write ("You lose") }                 NOT J
                                                 7
```

Mark G cannot be awarded due to faulty logic.
Marks H and J cannot be awarded as not all requirements have been met.
Ignore missing ;

VB.NET



VB.NET

```
d1 = int(9*rnd() +0)                                        PART C
d2 = int (9*rnd() +0)                                       PART C
dim over as boolean
dim guesses as Integer =0, correct as Integer =0
while not over                                              B, D
    correct =0
    guess =Console.readline()                               F
    guesses =guesses +1
    first = guess \ 10
    second = guess mod 10
    If d1 =first then
        correct +=1                                         PART G
    end if
    If d2 =second then
        correct +=1                                         PART G
    end if
    Console.write (correct)                                 NOT H
    If guesses =10 or correct =2 then
        over =true                                          E
    end if
end while
console.writeline("well done")                              NOT I
```

Mark A not awarded as d1 and d2 are not meaningful variable names. Mark J has not been attempted.

6

Get help and support

Visit our website for information, guidance, support and resources at [aqa.org.uk/8525](aqa.org.uk/8525)

You can talk directly to the Computer Science subject team

E: computerscience@aqa.org.uk

T: 0161 957 3980

aqa.org.uk