NAME:

Errors

There are three types of errors you can get in your code:

- Syntax
- Logic (semantic)
- Run Time

Explain each of these errors and state how they would manifest (make themselves known) to the programmer/user.

Syntax errors are about not using the correct use of language or punctuations which will display to the user as “syntax error”

a logic error is when something in your program is not realistic or add up it will work but not diplay the correct output.

Run time errors are when the program is linked to another source for things like images or styling but the reference to the computers is incorrect like the wrong https code link. It will simply fail to output anything as it will have no correct link to find it.

Errors in your Pseudocode/Flowchart Algorithm

Only one of the errors above can exist in your pseudocode/flowchart, which one is it and why?

A logic error as the program itself is right it's just not the correct calculation ect..

Planning

There are methods, things you can do to check your code before you even start to program. These are writing a **Dry Run** on paper by drawing a **Trace Table**. Basically, all you do is create a table with the variable names across the top and write in the values you would expect the variables to take through each line of the code.

Below is some pseudocode to create a list of the 2 times table and the trace table you would plan out to see if you had completed the logic correctly.

```
FOR index = 2 to 2
    print (“This is the “ + index “times table.”)
    FOR times = 2 to 12
        Print (“times + “X” + index + “ = “ + index * times)
    NEXT times
NEXT index
```

index	Output	times	Output
2	2		
2		2	4
2		3	6
2		4	8
2		5	10
2		6	12
2		7	14
2		8	16
2		9	18
2		10	20
2		11	22
2		12	24

Task 1

Look at the pseudocode below and complete the Trace Table.

```
turns
x = 3
WHILES turns < 22
        x = x * 3
        turns = turns + 3
END WHILE
print (x)
print (turns)
```

turns	x	Output
0	3	3
0	9	9
3	9	27

Task 2

Look at the pseudocode below and complete the Trace Table.

```
number_1 = 3
number_2 = 2
total = 0
FOR x = 1 to 5
        number_1 = number_1 * x
        number_2 = number_2 * (x + 1)
        total = number_1 + number_2
NEXT x
print (total)
```

number_1	number_2	total	Output
3	2	0	0
3	4	7	7
6	12	18	12
18	48		17

72	240	22	22
360	1440	27	27

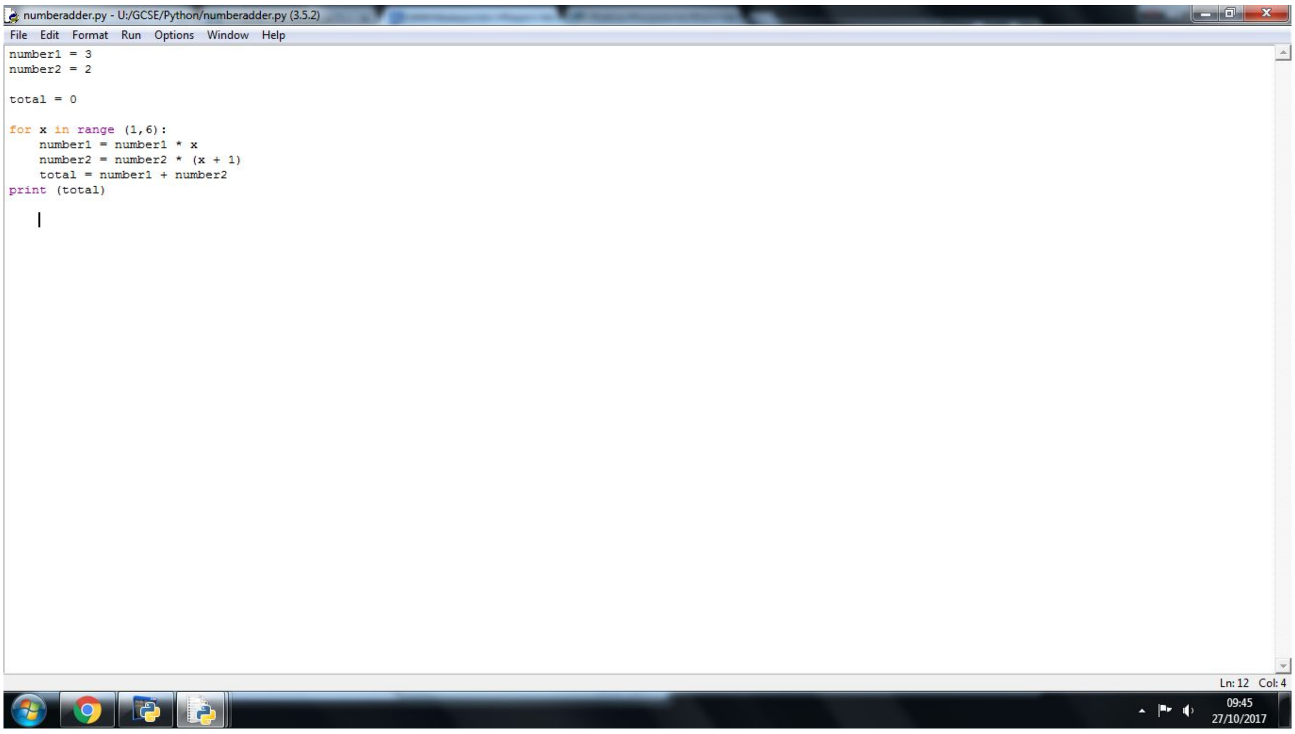
Task 3

Complete work from other programming or theory booklets that are overdue or deadline approaching.

Task 4

Program in Python the code for all three examples above:

- Two times table


numberadder.py - U:/GCSE/Python/numberadder.py (3.5.2)
File Edit Format Run Options Window Help
Ln: 12 Col: 4
09:45
27/10/2017

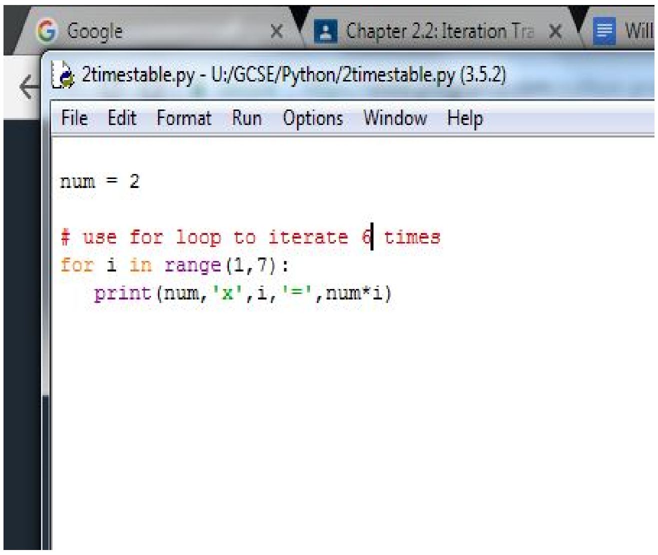
```
number1 = 3
number2 = 2

total = 0

for x in range (1,6):
    number1 = number1 * x
    number2 = number2 * (x + 1)
    total = number1 + number2
print (total)
```

- Turns

- number


Google
Chapter 2.2: Iteration Tra
Will
2timestable.py - U:/GCSE/Python/2timestable.py (3.5.2)
File Edit Format Run Options Window Help

```
num = 2

# use for loop to iterate 6 times
for i in range(1,7):
   print(num,'x',i,'=',num*i)
```