Lesson 5: Arithmetic expressions

Arithmetic expressions

```
number = 10-2+2*5
```

Question

What value will be held by the variable `number`?

1. -2
2. 18
3. 50
4. 30

Explorer task

Explain your answer

PEMDAS

Arithmetic expressions are evaluated in order of operator precedence.

You can use **PEMDAS** to help remind you of the correct order.

It is important to note that **add** and **subtract** are interchangeable and should be evaluated from left to right.

P – parentheses

E – exponents

M - multiplication

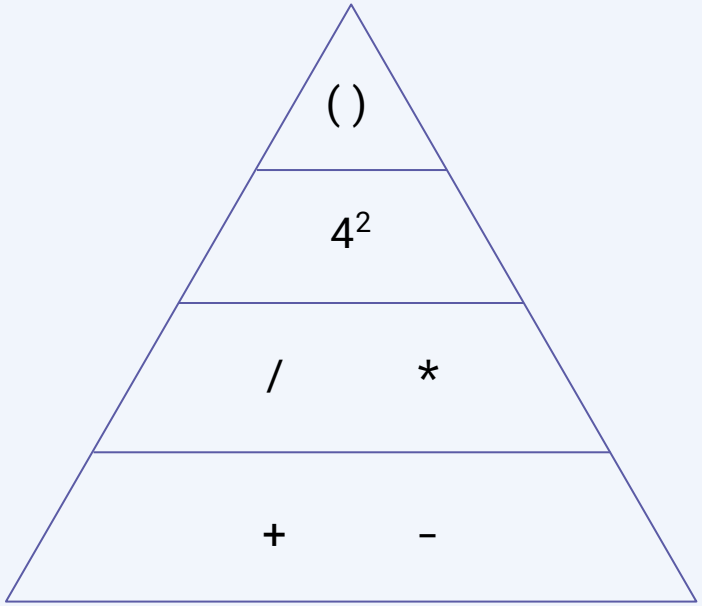
D - division

A - addition

S - subtract

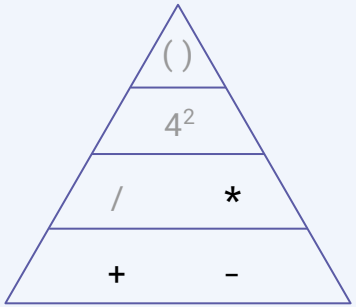
PEMDAS

In your **maths** lessons you might have seen **PEMDAS** represented in a triangle like this diagram.



PEMDAS

```
10-2+2*5
```

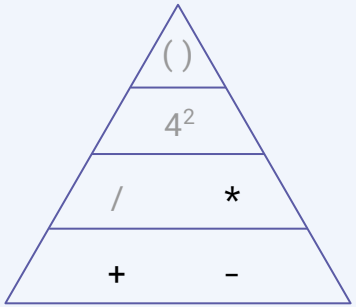


PEM**DAS**

There are no **p**arentheses, **e**xponents, or **d**ivision in this expression, so start with **m**ultiplication.

PEMDAS

`10-2+2*5`



PEM**DAS**

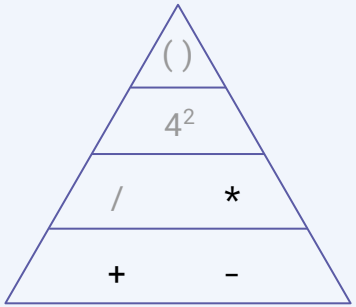
There are no **p**arentheses, **e**xponents, or **d**ivision in this expression, so start with **m**ultiplication.

`2*5 is 10`

PEMDAS

```
10-2+2*5

10-2+10
```



PEM**DAS**

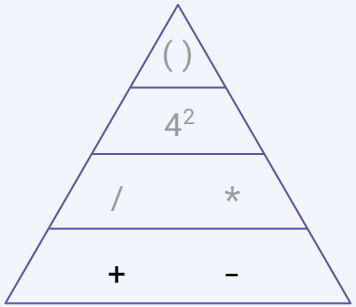
There are no **p**arentheses, **e**xponents, or **d**ivision in this expression, so start with **m**ultiplication.

`2*5 is 10`

PEMDAS

`10-2+2*5`

`10-2+10`



PEM**DAS**

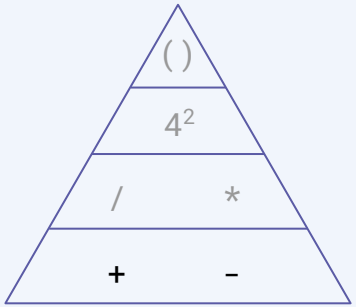
There are no **b**rackets, **i**ndices, or **d**ivision in this expression, so start with **m**ultiplication.

`2*5 is 10`

PEMDAS

`10-2+2*5`

`10-2``+10`



PEM**DAS**

Add and subtract should be read from left to right. If the subtract appears first, then this should be carried out first.

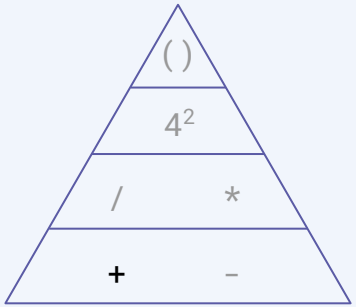
`10-2 is 8`

PEMDAS

10-2+2*5

10-2+10

8+10



PEMDAS

Add and subtract should be read from left to right. If the subtract appears first, then this should be carried out first.

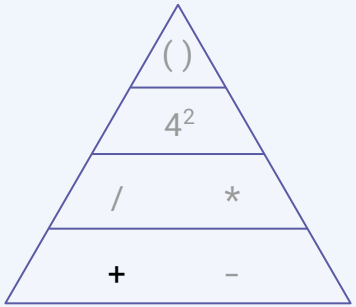
10-2 is 8

PEMDAS

```
10-2+2*5

10-2+10

8+10
```



PEMD**A**S

Finally, you are left with one **operator** and perform this final calculation.

```
8 + 10 is 18
```

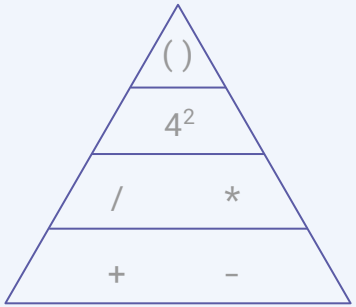
PEMDAS

```
10-2+2*5

10-2+10

8+10

18
```



PEMD**A**S

Finally, you are left with one **operator** and perform this final calculation.

```
8 + 10 is 18
```

Lesson 7: Arithmetic expressions

In this lesson, you will:

- Evaluate arithmetic expressions using rules of operator precedence (BIDMAS)
- Write and use expressions that use arithmetic operators (add, subtract, multiply, real division, integer division, MOD, to the power)
- Assign expressions to variables

Using operators in Python

Here is a list of **arithmetic operators** that can be used in Python.

+	Addition
-	Subtraction
*	Multiplication
/	Real division
//	Integer division (quotient)
**	Powers
%	Modulo (MOD)

Using operators in Python

Integer division and **modulo** are possibly unfamiliar to you.

+	Addition
-	Subtract
*	Multiplication
/	Real division
//	**Integer division (quotient)**
**	Powers
%	**Modulo (MOD)**

Using operators in Python

Here is an example of how **real division** will output.

In real division there is **no remainder** because the entire value is divided.

```
1  number = 14/3
2  print(number)
3
```

```
4.666666666666667
>>>
```

Using operators in Python

When you use **integer division,** it will discard the decimal part.

Integer division is the operation that calculates how many *whole* times the **divisor** (3) will fit in the **dividend** (14).

```
1  number = 14//3
2  print(number)
3
```

```
4
>>>
```

Using operators in Python

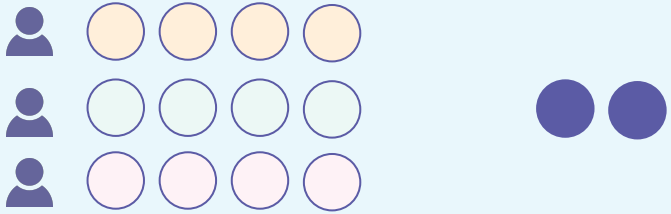
Modulo (MOD) is used to work out the remainder of the division.

```
1  number = 14%3
2  print(number)
3
```

```
2
>>>
```

Using operators in Python

If you divide **14** counters between **3** people, then there will be **4** counters each with **2** remaining.



```
1  counters_each = 14//3
2  counters_remaining = 14%3
3  print(counters_each)
4  print(counters_remaining)
```

```
4
2
>>>
```

Note: The MOD doesn't simply store the value that was discarded from integer division. It stores the whole number that was remaining.

Using operators in Python

You would use **modulo** when you need to find out the **remaining whole value**.

This will be helpful in many cases. An example might be when you wish to find out if a value is **odd** or **even**.

If the value evaluates as a **1**, then it is odd.

```
1  number = 27
2  odd_even = number%2
3  print(odd_even)
```

```
1
>>>
```

Using operators in Python

If the value evaluates as a `0`, then it is even.

```
1  number = 26
2  odd_even = number%2
3  print(odd_even)
```

```
0
>>>
```

Integer division and modulo

```
number = 25%4
```

Question

What value will be held by the variable `number`?

1. `6.25`
2. `1`
3. `6`
4. % is not a valid operator so this will cause an error message.

Integer division and modulo

```
number = 10/3
```

Question

What value will be held by the variable `number`?

1. `3.3333333333333335`
2. 3
3. 1
4. / is not a valid operator so this will cause an error message.

Integer division and modulo

```
number = 13//3
```

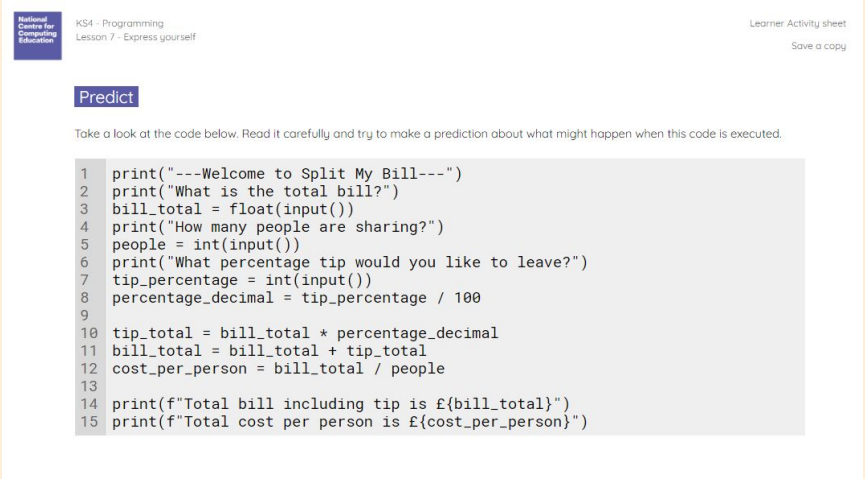
Question

What value will be held by the variable `number`?

1. 1
2. `4.333333333333333`
3. 4
4. // is not a valid operator so this will cause an error message.

The ‘Split my bill’ app

Use the **‘Split my bill’ worksheet** to investigate a program that has been designed to help friends split the bill at a restaurant.



National Centre for Computing Education

KS4 - Programming
Lesson 7 - Express yourself

Learner Activity sheet
Save a copy

Predict

Take a look at the code below. Read it carefully and try to make a prediction about what might happen when this code is executed.

```
print("---Welcome to Split My Bill---")
print("What is the total bill?")
bill_total = float(input())
print("How many people are sharing?")
people = int(input())
print("What percentage tip would you like to leave?")
tip_percentage = int(input())
percentage_decimal = tip_percentage / 100

tip_total = bill_total * percentage_decimal
bill_total = bill_total + tip_total
cost_per_person = bill_total / people

print(f"Total bill including tip is £{bill_total}")
print(f"Total cost per person is £{cost_per_person}")
```

Learning partner feedback

Discuss your solution to the **make** task with your learning partner.

- Did you meet the requirements?
- Did you try the explorer task?
- What errors did you encounter and how did you overcome them?
- How could you improve your program?

Homework: Finish the app

Complete the ‘Split my Pizza’ app.

Make improvements based on your peer feedback.

Due: Next lesson

Next lesson

In this lesson, you...

Learnt about BIDMAS and how this can help you with order of precedence

Explored arithmetic expressions

Familiarised yourself with integer division and modulo

Next lesson, you will...

Learn about selection statements and how these can be used to control the flow of execution in programs