Arithmetic, comparison and Boolean operators

Algorithms and programming (Paper 2) · revision notes

Operators let a program calculate, compare and combine conditions. You need the arithmetic operators (including MOD, DIV and ^), the comparison operators and the Boolean operators AND, OR and NOT. Expect 'evaluate this expression' questions and conditions inside code you write.

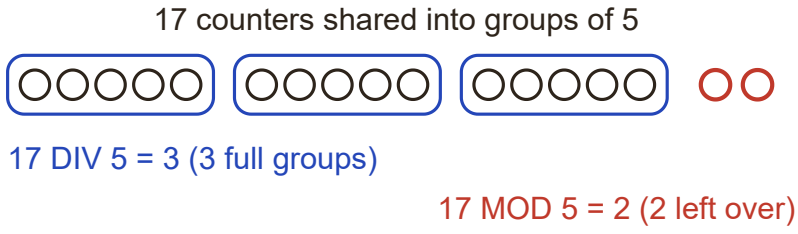
Key facts

- + − * / : add, subtract, multiply, divide (/ can give a decimal)
- DIV = whole-number division: 17 DIV 5 = 3
- == equal, != not equal, < <= > >= compare values (result True or False)
- Even number test: num MOD 2 == 0
- MOD = remainder: 17 MOD 5 = 2
- ^ = power: 3 ^ 2 = 9
- AND: both true; OR: at least one true; NOT: reverses
- Inside a range: AND; outside a range: OR

Notes

Arithmetic operators

- `+` add, `-` subtract, `*` multiply, `/` divide (7 / 2 = 3.5).
- `MOD` gives the **remainder** of a division: 17 MOD 5 = 2.
- `DIV` gives the **whole-number part** (quotient) of a division: 17 DIV 5 = 3.



DIV counts the full groups; MOD is what is left over

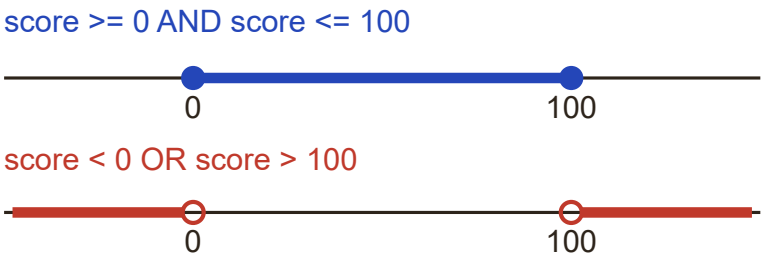
- `^` raises to a power (exponent): 2 ^ 3 = 8.
- Brackets are worked out first, then powers, then * / MOD DIV, then + and −. Use brackets to make the order clear.
- Uses: `num MOD 2 == 0` is true for even numbers; 135 seconds is 135 DIV 60 = 2 minutes and 135 MOD 60 = 15 seconds.

Comparison operators

- `==` equal to, `!=` not equal to, `<` less than, `<=` less than or equal to, `>` greater than, `>=` greater than or equal to.
- A comparison always gives a Boolean result: True or False.
- Check boundaries carefully: `age > 18` is false when age is 18; `age >= 18` is true.

Boolean operators

- `AND`: True only if **both** conditions are True.
- `OR`: True if **at least one** condition is True.
- `NOT`: reverses the result, so NOT True is False.
- A range check needs AND: `score >= 0 AND score <= 100`. Outside a range needs OR: `score < 0 OR score > 100`.



Inside a range: AND. Outside a range: OR

- Each side of AND or OR must be a full comparison: write `x == 1 OR x == 2`, not `x == 1 OR 2`.

How to answer each type of question

Evaluate expressions 1 mark each **Grade 5**

1. Do brackets first, then powers, then * / MOD DIV, then + and −.
2. For MOD, find the remainder; for DIV, the whole number of times it goes in.
3. For a Boolean expression, work out each comparison as True or False first, then apply AND, OR or NOT.

Example. State the result of each expression.
(a) 23 MOD 4 (b) 23 DIV 4 (c) 3 ^ 2 + 1 (d) (10 > 3) AND (4 == 5) [4]

Model answer.

(a) 3 (1)

(b) 5 (1)

(c) 10 (1)

(d) False (1): 10 > 3 is True but 4 == 5 is False, and AND needs both.

Write a condition

2 marks **Grade 6**

1. Decide which comparisons are needed, and whether the limits are included (use <= or >=).
2. Join them with AND if all must be true, or OR if any one is enough.
3. Write each comparison in full.

Example. Write a condition that is True when `age` is from 13 to 19 inclusive and `member` is True. [2]

Model answer.

```
age >= 13 AND age <= 19 AND member == true
```

Correct range 13 to 19 inclusive using AND (1); joined with AND to member being True (1).

Use MOD and DIV in a program

3 marks **Grade 7**

1. Use DIV to find how many whole units fit.
2. Use MOD to find what is left over.
3. Cast numbers with str() when you join them into a message.

Example. Write code that inputs a number of seconds and outputs it in minutes and seconds. For example, 200 should output `3 minutes 20 seconds`. [3]

Model answer.

```
secs = int(input("Enter seconds"))
mins = secs DIV 60
left = secs MOD 60
print(str(mins) + " minutes " + str(left) + " seconds")
```

Uses DIV 60 to find the minutes (1); uses MOD 60 to find the seconds left over (1); outputs both values in a message (1).

Don’t lose marks

- Swapping MOD and DIV.
- Using > when the question says 'or more' (that needs >=).
- Writing `x == 1 OR 2` instead of `x == 1 OR x == 2`.
- Using AND for 'outside a range', which can never be true (a number cannot be below 0 and above 100 at once).
- Using / when a whole-number answer is needed: 7 / 2 is 3.5, but 7 DIV 2 is 3.

More tips

Memory tricks

- MOD = 'what's left over' (remainder).
  DIV = 'how many whole times it goes in'.
- Check: (a DIV b) × b + (a MOD b) should equal a. For 17 and 5: 3 × 5 + 2 = 17.
- AND is strict (both), OR is generous (either).

Exam technique

- Show the working for MOD and DIV (for example 23 = 5 × 4 + 3) so a slip can still earn credit.
- Words to watch: 'at least' and 'or more' mean >=; 'at most' and 'up to' mean <=; 'more than' means >.
- In Boolean questions, write True or False for each part before combining them.

What each grade needs

What you need to be able to do, from the first marks up to the top grade.

Grade 3 — **Use the basic arithmetic operators**
Add, subtract, multiply and divide with +, −, * and /.

Grade 4 — **Use the comparison operators correctly**
Use ==, !=, <, <=, > and >= to compare values and give True or False.

Grade 5 — **Evaluate expressions using MOD, DIV and ^**
17 MOD 5 = 2, 17 DIV 5 = 3 and 2 ^ 3 = 8.

Grade 5 — **Combine conditions with AND, OR, NOT**
Work out whether a combined condition is True or False.

Grade 6 — **Write a condition for a range**
For 13 to 19 inclusive: age >= 13 AND age <= 19.

Grade 7 — **Use MOD and DIV to solve problems**
Test for even numbers, or split a total into units, such as minutes and seconds.

Free notes, Full packs, practice papers and mock exams at chhetriacademy.com. Written for this site; not taken from past papers. Stuck? 1-to-1 tutoring with a free 30-minute trial lesson.