

Exercise walk-through

2.1

Algorithms with Selection and Repetition

AP Computer Science A

Boolean expressions & the if statement

A **Boolean expression**

operators

evaluates to true or false using **relational**
($>$, $<$, $==$, $!=$).

if (cond)
run a block

if / else
pick one of two

else if
three or more paths

This ability to choose a path is **selection**. Trace which **branch** runs for a given input.



Assembly loop

The three families of operators

arithmetic

+ **-** ***** **/**

MOD **DIV**

give a
number

relational

= **<>** **<** **>**

<= **>=**

compare →
true or **false**

logical

AND **OR**

NOT

join conditions →
true or **false**

The three families of operators: arithmetic, relational, and logical

Compound & nested conditions

&& (AND)
both true

|| (OR)
at least one

! (NOT)
reverse

Short-circuit evaluation stops early: a && b skips b if a is false. Compare **objects** with `.equals()`, not `==` (which compares references).



Traffic light if

Nested and chained conditions: only the first match runs

nesting

an if inside another – the inner one is reached only when the outer condition was true

chaining

else if tests several cases **in order**

the rule

only the first matching branch runs; the rest are skipped even if they also match

so order matters

put the **most specific** test first – score ≥ 50 before score ≥ 80 gives everyone the lower grade

else

catches everything no earlier branch matched – it needs no condition of its own

Tracing a chain means reading **top to bottom and stopping at the first true**. That is the whole semantics, and most multiple-choice items test only this.

Exercises 2.1-2.3

2.1 Name the three building blocks of an algorithm.

[2]

2.2 State what selection lets an algorithm do.

[1]

2.3 State what repetition lets an algorithm do.

[1]

Exercises 2.1-2.3

2.1 Name the three building blocks of an algorithm. [2]

2.1

- sequencing, selection, repetition
-

2.2 State what selection lets an algorithm do. [1]

2.2

- it lets the algorithm choose between different paths based on a condition
-

2.3 State what repetition lets an algorithm do. [1]

2.3

- it lets the algorithm repeat a block of steps

Exercises 3.1-3.2

3.1 Choosing between paths in an algorithm is

[1]

☐

A sequencing

B selection

C repetition

D casting

3.2 Repeating a block of steps is

[1]

☐

A sequencing

B selection

C repetition

D overloading

Exercises 3.1-3.2

3.1 Choosing between paths in an algorithm is

[1]

☐

A sequencing

B selection

C repetition

D casting

3.1 B

- selection.

3.2 Repeating a block of steps is

[1]

☐

A sequencing

B selection

C repetition

D overloading

3.2 C

- repetition.

Exercise 3.3 (a)

3.3 An algorithm checks each of 10 scores and prints those above 50.

(a) Name the building block that repeats the check.

[1]

Exercise 3.3 (a)

3.3 An algorithm checks each of 10 scores and prints those above 50.

(a) Name the building block that repeats the check.

[1]

(a)

- repetition

Exercise 3.3 (b)

3.3 An algorithm checks each of 10 scores and prints those above 50.

(b) Name the building block that decides "above 50".

[1]

Exercise 3.3 (b)

3.3 An algorithm checks each of 10 scores and prints those above 50.

(b) Name the building block that decides "above 50".

[1]

(b)

- selection

Exercise 3.3 (c)

3.3 An algorithm checks each of 10 scores and prints those above 50.

(c) Name the building block that runs steps in order.

[1]

Exercise 3.3 (c)

3.3 An algorithm checks each of 10 scores and prints those above 50.

(c) Name the building block that runs steps in order.

[1]

(c)

- sequencing