GRADING ENVIRONMENT (TEST-BED) (40/40 points)

This is a dummy exercise—the same one than in the associated video—so you might experiment with the exercise environment. The resulting grade will not be taken into account in the final grading.

Even if you don't know any OCaml yet, the editor below has been initialised with trivial and almost valid OCaml code, that partially answers the exercise. You might press the `Check & Save` button in order to get a **sample grading report**.

Submission and saving

By trial and guess, you might edit the answer and achieve the maximum grade. At any time, if you feel lost, you might **revert the editor** to the default template by pressing `Reset Template`. If you pressed this button too quickly, you might undo the modification with `Ctrl-Z`, or `Command-Z` on Apple keyboards.

You might also revert the editor to the **last saved answer** by reloading the webpage. Your answer is saved every time you press the `Check & Save` button, but also when you press the `Check` button below the current grading report. You might **check-save-grade** as many times as you wish. Only the last submitted solution will be taken into account.

Error message

If the automatic grader rejects the solution because of a syntax or a typing error, the editor will be **annotated with located error messages**. Just pass the mouse over the left red cross to get the detailed error messages. You might also get the exact same error messages whitout saving your solution by pressing the `Typecheck` button. We advise you to typecheck your code before submitting to the grader. Typing errors tends to be extremely precise—may be too much for beginners—while syntax error tends to be quite terse. Don't hesitate to **ask questions** on the forum, if you can not make sense of it yet.

Forced-indentation

OCaml does not have an indentation-dependent syntax. You are usually free not to indent your code—even if nobody here will ever advise you to do so. However, for this MOOC, the online editor is configured by default with **forced-indentation mode**. Don't be surprised if your code moves around while you are typing, the current editor might react and reindent a bit too promptly. If you are puzzled by the indentation chosen for you, well, the editor is probably right and you might have made a syntax error.

Toplevel

You might also experiment with the **OCaml toplevel** by pressing `Evaluate >>>` or `Switch >>>`. The former will reset the OCaml toplevel and try to load the content of the editor into it; while the latter will simply switch view, leaving the toplevel in its current state.

THE TASK

In this test-bed exercise you are asked to (re)implement the basic integer-arithmetic functions.

1. Write a function `plus` of type `int -> int -> int`.
2. Write a function `minus` of type `int -> int -> int`.
3. Write a function `times` of type `int -> int -> int`.
4. Write a function `divide` of type `int -> int -> int`.

The preloaded template contains a minor syntax error and is only a partially valid answer. These errors were introduced in order to let you experiment with the error reporting mechanism and the grading report.
Feel free to introduce more errors and to stress the system, the resulting grade for this exercise will not be taken into account in the global grade and you might submit as many solutions as you wish.
If you end up writing an infinite computation, the system will detect it after a while and ask you

to stop the script. It will slow your browser down until that point, since everything is done on your side, via your JavaScript engine. So don't worry, you can try and break the system as much as you want, it should not break anything on our servers.

THE GIVEN PRELUDE

```
(* Some code is loaded in the toplevel before your code. *)

let greetings = "Hello world!"
```

YOUR OCAML ENVIRONMENT

```

let plus x y = x + y ;;

let minus x y = x - y ;;

let times x y = x * y ;;

let divide x y = x / y ;;

```

Evaluate >

Switch >:

Typechecl

Reset Templ

Full-screen |

Check & Sa

```
Exercise complete (click for details)                          40 pts
v Function: plus                                     Completed, 10 pts
  Found plus with compatible type.
  Computing plus 1 1
  Correct value 2                                                1 pt
  Computing plus 2 2
  Correct value 4                                                1 pt
  Computing plus 10 -10
  Correct value 0                                                1 pt
  Computing plus -4 3
  Correct value -1                                               1 pt
  Computing plus -3 -4
  Correct value -7                                               1 pt
  Computing plus -4 2
  Correct value -2                                               1 pt
  Computing plus 4 4
  Correct value 8                                                1 pt
  Computing plus -3 3
  Correct value 0                                                1 pt
  Computing plus 1 -4
  Correct value -3                                               1 pt
  Computing plus 0 -4
  Correct value -4                                               1 pt
v Function: minus                                    Completed, 10 pts
  Found minus with compatible type.
  Computing minus 1 1
  Correct value 0                                                1 pt
  Computing minus 4 -2
  Correct value 6                                                1 pt
  Computing minus 0 10
  Correct value -10                                              1 pt
```

```
Computing minus 4 -5
Correct value 9                                                    1 pt
Computing minus 2 -3
Correct value 5                                                    1 pt
Computing minus 2 4
Correct value -2                                                   1 pt
Computing minus -1 -4
Correct value 3                                                    1 pt
Computing minus 0 4
Correct value -4                                                   1 pt
Computing minus -2 -3
Correct value 1                                                    1 pt
Computing minus -4 -2
Correct value -2                                                   1 pt
v Function: times                                  Completed, 10 pts
Found times with compatible type.
Computing times 1 3
Correct value 3                                                    1 pt
```

```
Correct value 0                                                    1 pt
Computing times 4 1
Correct value 4                                                    1 pt
Computing times -2 -4
Correct value 8                                                    1 pt
Computing times -4 -4
Correct value 16                                                   1 pt
Computing times -4 -5
Correct value 20                                                   1 pt
Computing times 3 3
Correct value 9                                                    1 pt
Computing times -3 -1
Correct value 3                                                    1 pt
Computing times -3 2
Correct value -6                                                   1 pt
v Function: divide                                 Completed, 10 pts
Found divide with compatible type.
Computing divide 12 4
Correct value 3                                                    1 pt
Computing divide 12 5
Correct value 2                                                    1 pt
Computing divide 3 0
Correct exception Division_by_zero                                 1 pt
Computing divide -1 3
Correct value 0                                                    1 pt
Computing divide 2 -2
Correct value -1                                                   1 pt
Computing divide -3 3
Correct value -1                                                   1 pt
Computing divide 1 -2
Correct value 0                                                    1 pt
Computing divide 4 -1
Correct value -4                                                   1 pt
Computing divide -3 2
Correct value -1                                                   1 pt
Computing divide 4 1
Correct value 4                                                    1 pt
```