**Table of Contents**

## STRING EXPRESSIONS (5/5 points)

What is the result of compiling and evaluating `"12" + "34"` ?

○ Syntax error.

● Type error. ✓

○ `46`

○ `1234`

○ `"1234"`

The **string-concatenation** operator is `^`. The `+` operator is only the integer addition.

What is the result of compiling and evaluating `"12" ^ (string_of_int 3) ^ "45"` ?

○ Syntax error.

○ Type error.

○ `12345`

● `"12345"` ✓

- The `string_of_int` function pretty-prints an integer into a string. The evaluation of `(string_of_int 3)` is then the string `"3"`

- Exceptions, input/output and imperative constructs
- Modules and data abstraction

- String literals are always written between double quotes, even by the generic printer included in the OCaml toplevel.

What is the result of compiling and evaluating `"12" ^ string_of_int 3 ^ "45"`?

○ Syntax error.

○ Type error.

○ `12345`

● `"12345"` ✓

Function application has a higher priority than the basic infix operators ( `+` , `*.` , `^` , `mod` , ...). Then, this expression is read: `"12" ^ (string_of_int 3) ^ "45"`

What is the result of compiling and evaluating `"She texted me: "hi :-)"!!!"`?

● Syntax error. ✓

○ Type error.

○ `"She texted me: "hi :-)"!!!"`

- As usual, special characters in string needs to be **escaped**, e.g. in this particular string the `"` character.
- A correct string constant could be `"She texted me: \"hi :-)\"!!!"`

What is the result of compiling and evaluating `String.get "abcd" 1`?

○ Syntax error.

○ Type error.

○ `'a'`

○ `a`

● `'b'` ✓

○ `b`

- Quoting the OCaml manual: "`String.get s n` returns the character at index `n` in string `s`".
- As usual, character indexes in strings **starts at 0**.
- **Character literals** are written between single quotes: for instance `'a'`, `'A'` or `' '`.

*Vous avez utilisé 1 essais sur 3*

## STRING EXPRESSIONS (BIS) (4/4 points)

**Warning:** you only have 1 attempt (but anyway the result will not count in the final grading).

What is the result of compiling and evaluating `"12" ^ "34"`?

○ Syntax error.

○ Type error.

○ `46`

○ `1234`

● `"1234"` ✓

The `^` operator is the string concatenation.

What is the result of compiling and evaluating `12 + "34"`?

○ Syntax error.

● Type error. ✓

○ `46`

○ `1234`

○ `"1234"`

No implicit conversion between types in OCaml.

What is the result of compiling and evaluating `"She texted me: \"hi :-)\"!!!"`?

○ Syntax error.

○ Type error.

○ `She texted me: "hi :-)"!!!`

● `"She texted me: \"hi :-)\"!!!"` ✓

This string literal is properly escaped.

What is the result of compiling and evaluating `String.get "zyxw" 2`?

○ Syntax error.

○ Type error.

○ `'z'`

○ `'y'`

● `'x'` ✓

○ `'w'`

Character indexes in strings starts at 0.

*Vous avez utilisé 1 essais sur 1*

## CONVERSION FUNCTIONS (3/3 points)

What is the result of compiling and evaluating `(int_of_string "12") + 3` ?

○ Syntax error.

○ Type error.

● `15` ✓

○ `"123"`

- The `int_of_string` function returns the integer value read in its string argument. Or fails, if its argument does not represent an integer.
- The sub-expression `(int_of_string "12")` evaluates into the integer value of `12`.

What is the result of compiling and evaluating `float_of_string "12.5" +. 2.5`?

○ Syntax error.

○ Type error.

● `15.` ✓

○ `"12.52.5"`

Function application has greater priority than the basic infix operators like `+.`; the expression is read `(float_of_string "12.5") +. 2.5`. It is a valid floating-point expression.

What replacement for the function `f` renders the expression `f 3. ^ "1415"` correct?

○ `string_of_int`

● `string_of_float` ✓

○ `int_of_float`

○ `int_of_string`

○ `float_of_int`

○ `float_of_string`

- This expression is read as: `(f 3.) ^ "1415"`.

- For the expression to be valid, the sub-expression `(f 3.)` should be of type `string`.
- The literal `3.` being a float literal, the `f` function should be a function that receives a `float` and returns a `string`.

*Vous avez utilisé 1 essais sur 3*

## CONVERSION FUNCTIONS (BIS) (4/4 points)

**Warning:** you only have 1 attempt (but anyway the result will not count in the final grading).

What replacement for the function `f` renders the expression `f "3" + 1415` correct?

○ `string_of_int`

○ `string_of_float`

○ `int_of_float`

● `int_of_string` ✓

○ `float_of_int`

○ `float_of_string`

This is the integer addition and `"3"` is a string literal.

What replacement for the function `f` renders the expression `f 3 ^ "1415"` correct?

● `string_of_int` ✓

○ `string_of_float`

○ `int_of_float`

○ `int_of_string`

○ `float_of_int`

○ `float_of_string`

This is a string concatenation and `3` is an integer literal.

What replacement for the function `f` renders the expression `f 3 +. 1415.` correct?

○ `string_of_int`

○ `string_of_float`

○ `int_of_float`

○ `int_of_string`

● `float_of_int` ✓

○ `float_of_string`

This is a floating-point addition and `3` is an integer literal.

What replacement for the function `f` renders the expression `f 3 < "1415"` correct?

● `string_of_int` ✓

○ `string_of_float`

○ `int_of_float`

○ `int_of_string`

○ `float_of_int`

○ `float_of_string`

This is a polymorphic comparison against a string and `3` is an integer literal.

*Vous avez utilisé 1 essais sur 1*

## COMPARISON EXPRESSIONS (3/3 points)

What is the result of compiling and evaluating `"abcd" = ("ab" ^ "cd")`?

○ Syntax error.

○ Type error.

○ `false`

● `true` ✓

Polymorphic comparison operators are able to compare strings.

What is the result of compiling and evaluating `"a" ^ "bcd" = "ab" ^ "cd"`?

○ Syntax error.

○ Type error.

○ `false`

● `true` ✓

The concatenation operator `^` has a greater priority than the equality operator `=`. This expression is read as `("a" ^ "bcd") = ("ab" ^ "cd")` and both sub-expressions evaluate to `"abcd"`.

What is the result of compiling and evaluating `"Alphabet" < "Gammadelt"`?

○ Syntax error.

○ Type error.

○ `false`

● `true` ✓

The comparison of strings is based on the (ASCII-)alphanumerical order.

*Vous avez utilisé 1 essais sur 3*

## BOOLEAN EXPRESSIONS (3/3 points)

**Warning:** you only have 1 attempt (but anyway the result will not count in the final grading).

What is the result of compiling and evaluating `"Aa" < "Ob" && "Wagon" <> "Zephyr"`?

○ Syntax error.

○ Type error.

○ `false`

The comparison of strings is based on the (ASCII-)alphanumerical order.

What is the result of compiling and evaluating `12 <= "34" || 56 <= 78` ?

○ Syntax error.

● Type error. ✓

○ `false`

○ `true`

You can't compare integer and string without explicit conversion of one of the arguments.

What is the result of compiling and evaluating `not ("12" <= "34" || 56 <= 78)` ?

○ Syntax error.

○ Type error.

● `false` ✓

○ `true`

The two comparisons are true.

*Vous avez utilisé 1 essais sur 1*

Rechercher un cours

▼

A propos

Aide

Contact

Conditions générales d'utilisation

Charte utilisateurs

Politique de confidentialité

Mentions légales

POWERED BY OPENedX