PA2 - OCaml Practice

Programming Languages (SWE3006-41)
Spring, 2024

Instructor :
Sungjae Hwang
- jason.sungjae.hwang@gmail.com
- https://softsec-lab.github.io

TA :
- Kyeonghwan Min

Software
Security Lab
SUNGKYUNKWAN UNIV

Introduction

- Deadline : 2023/04/30 (Delay Submission 2024/05/02, 25% deduction per day.)

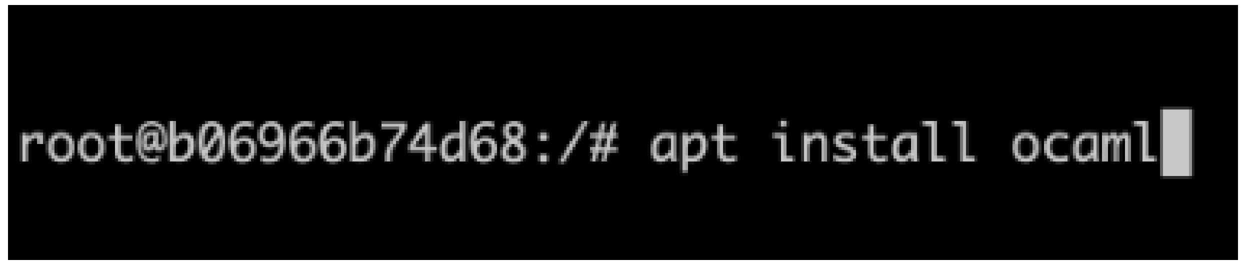
- Write functions using OCaml.

- Submit source codes (*.ml) for each exercise.
  - You will not get any points if your source code does not compiles well.
  - Submit "PA2_OCaml_StudentID.zip" through icampus.
  - The zip file should contains :
  -> ex1.ml, ex2.ml, ex3.ml, ex4.ml, ex5.ml, ex6.ml, ex7.ml

- Please leave the questions in the google sheet.
  https://docs.google.com/spreadsheets/d/1ncqaTXNTBvwoK0QltBKkY0r9jcxwyx7-gzCObOhKZuc/edit#gid=259444581
  * Avoid using email or the iCampus message for inquires.

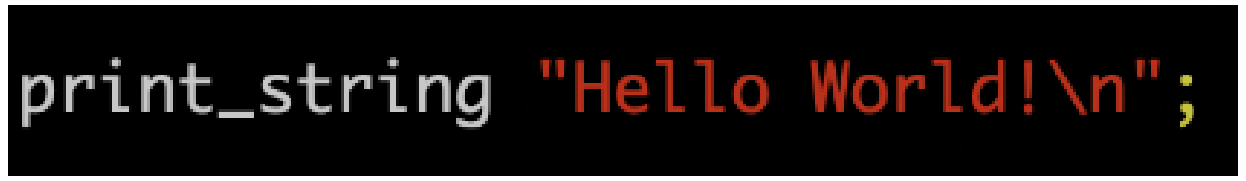
Installing OCaml

- For your information : https://ocaml.org/docs/install.html
- Hello World Example (Linux)



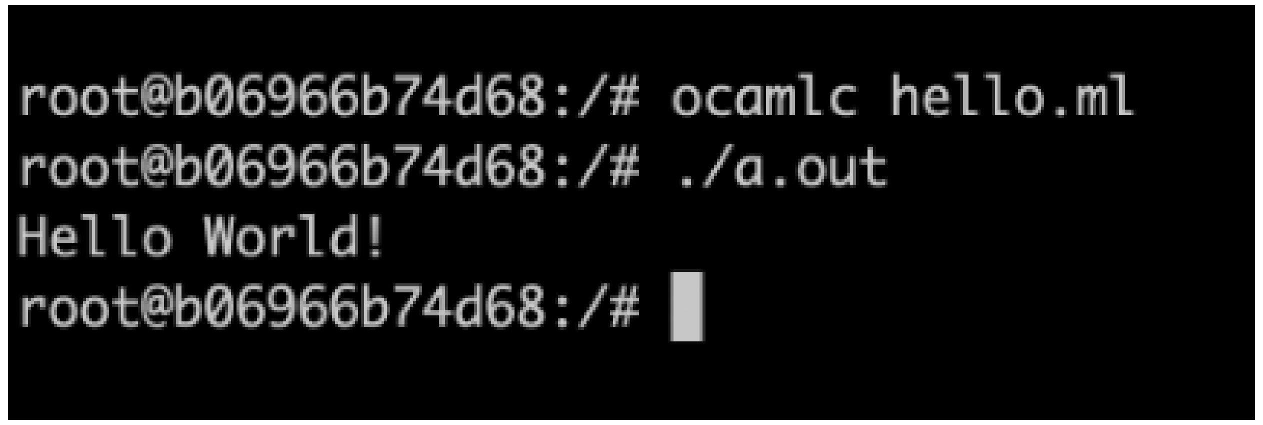
```
root@b06966b74d68:/# apt install ocaml
```

Installing OCaml



```
print_string "Hello World!\n";
```

hello.ml file



```
root@b06966b74d68:/# ocamlc hello.ml
root@b06966b74d68:/# ./a.out
Hello World!
root@b06966b74d68:/#
```

Compling and running

Exercise #1 (5pt)

- Write below function
  - gcd : int -> int -> int
  - The function returns the greatest common divisor(GCD) of two given non-negative integers.
  - Use the Euclidean algorithm based on following principle (n, m are integer that n>=m) :

$$\text{gcd } n\ m = \begin{cases} n & (m = 0) \\ \text{gcd } (n - m)\ m & \end{cases}$$

- Test Cases
  - gcd 10 0 => 10
  - gcd 9 5 => 1
  - gcd 13 13 => 13
  - gcd 37 600 => 1
  - gcd 0 0 => 0

Exercise #2 (10pt)

- Write below function
  - palindrome: 'a list -> bool
  - Check if given list is palinedrome.

- Test Cases
  - palindrome ["1"; "2"; "3"; "4"] => false
  - palindrome ["x"; "m"; "a"; "s"] => false
  - palindrome ["a"; "m"; "o"; "r"; "e"; "r"; "o"; "m"; "a"] => true
  - palindrome ["1"; "2"; "3"; "2"; "1"] => true
  - palindrome ["b"; "o"; "r"; "r"; "o"; "w"; "o"; "r"; "r"; "o"; "b"] => true

Exercise #3 (10pt)

- Write below function
  - factor_list : int -> int -> (int * int) list
  - Print the list of factors of given number N.
  - Each tuple means : (factor, the number of factor)

- Test Cases
  - fibo 10 => [(2, 1); (5, 1)]
  - fibo 17 => [(17, 1)]
  - fibo 27 => [(3, 3)]
  - fibo 315 => [(3, 2); (5, 1); (7, 1)]
  - fibo 777 => [(3, 1); (7, 1); (37, 1)]
  - fibo 1024 => [(2, 10)]

Exercise #4 (15pt)

- Write below function
  - phi : int -> int
  - The function returns the number of positive integers r that are coprime to m.
  - The range of positive integers r is $1 \leq r < m$
  - Let $\varphi(1) = 1$

- Test Cases
  - phi 4 => 2
  - phi 9 => 6
  - phi 10 => 4
  - phi 17 => 16
  - phi 30 => 8

Exercise #5 (20pt)

- Write below function
  - goldbach_list_limit : int -> int -> int -> (int * (int * int)) list
  - The function returns a list of goldbach composition given lower and upper limit.
  - N is a lower limit of each element of a goldbach composition.
  - If there are multiple cases in a number, only consider the composition has the smallest number and check if it is bigger or equal than limit.
- Test Cases
  - goldbach_list_limit 9 20 5 => [(12, (5, 7)); (18, (5, 13))]
  - goldbach_list_limit 25 70 10 => []
  - goldbach_list_limit 100 100 100 => []
  - goldbach_list_limit 100 200 19 => [(128, (19, 109))]
  - goldbach_list_limit 50 500 20 => [(220, (23, 197)); (308, (31, 277)); (346, (29, 317)); (488, (31, 457))]
  - goldbach_list_limit 1 2000 50 => [(992, (73, 919)); (1382, (61, 1321)); (1856, (67, 1789)); (1928, (61, 1867))]

Exercise #6 (20pt)

- Write below function
  - sigma : int * int * (int -> int) -> int
  - Such that sigma(a, b, f) returns as follow :

$$\sum_{n=a}^{b} f(n)$$

- Test Cases
  - sigma (10, 10, (fun x -> x)) => 10
  - sigma (11, 10, (fun x -> x)) => 0
  - sigma (10, 5, (fun x -> x)) => 0
  - sigma (1, 10, (fun x -> if x mod 2 = 0 then 1 else 0)) => 5
  - sigma (2, 10, (fun x -> x + 10)) => 144
  - sigma (0, 100, (fun x -> 0)) => 0
  - sigma (10, 12, (fun x -> 2 * x)) => 66

Exercise #7 (20pt)

- In OCaml, there is a function "fold" for lists :
  - fold : ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a
  - Recombines then results of recursively processing its constituent parts, building up a return value through use of combining operation.
  - For example, fold f a [b1;b2;...;bn] = f (...(f (f a b1) b2) ...) bn.

- Extend fold function so that it takes three lists. Write below function :
  - fold3 : ('a -> 'b -> 'c -> 'd -> 'a) -> 'a -> 'b list -> 'c list -> 'd list -> 'a
  - of which means,
  - fold3 f a [b1;b2;...;bn] [c1;c2;...;cn] [d1;d2;...dn] = f (...(f (f a b1 c1 d1) b2 c2 d2)...) bn cn dn.
  - You may assume that all the given lists are of the same length.

- Test Cases
  - fold3 (fun a b c d -> a + b + c + d) 10 [33;67;12;33] [10;23;84;57] [11;55;23;58] => 476
  - fold3 (fun a b c d -> (-a) + b + c + d) 4 [11;63;-45;22] [75;123;-44;1] [55;24;20;3] => 168
  - fold3 (fun a b c d -> a * b * c * d) 55 [] [] [] => 55
  - fold3 (fun a b c d -> (a * b * c + d) mod 7) 33 [12;33] [10;7] [5;12] => 5
  - fold3 (fun a b c d -> if b then a + c else a + d) 34 [true;false;false;true] [12;3;4;77] [11;23;6;100] => 152
  - fold3 (fun a b c d -> if b then a else c + d) 55 [true;true;false;false;true] [111;63;88;123;98] [0;23;778;34;6] => 157

Grading

- Run the below command with the given testcase file, "testcase.txt"
  `diff <(seq 7 | xargs -I % ocaml ex%.ml) testcase.txt`

- If there are no output in command line, that means you could get full score.

- Asking about your grade without checking your output would not be considered in any way.

- Any plagiarism detected would be get 0 points and would lead you to F.

Additional Information

- TA will just compile and execute your file. So, there are no inputs.

- All testcases are written in pdf. No more additional testcases.

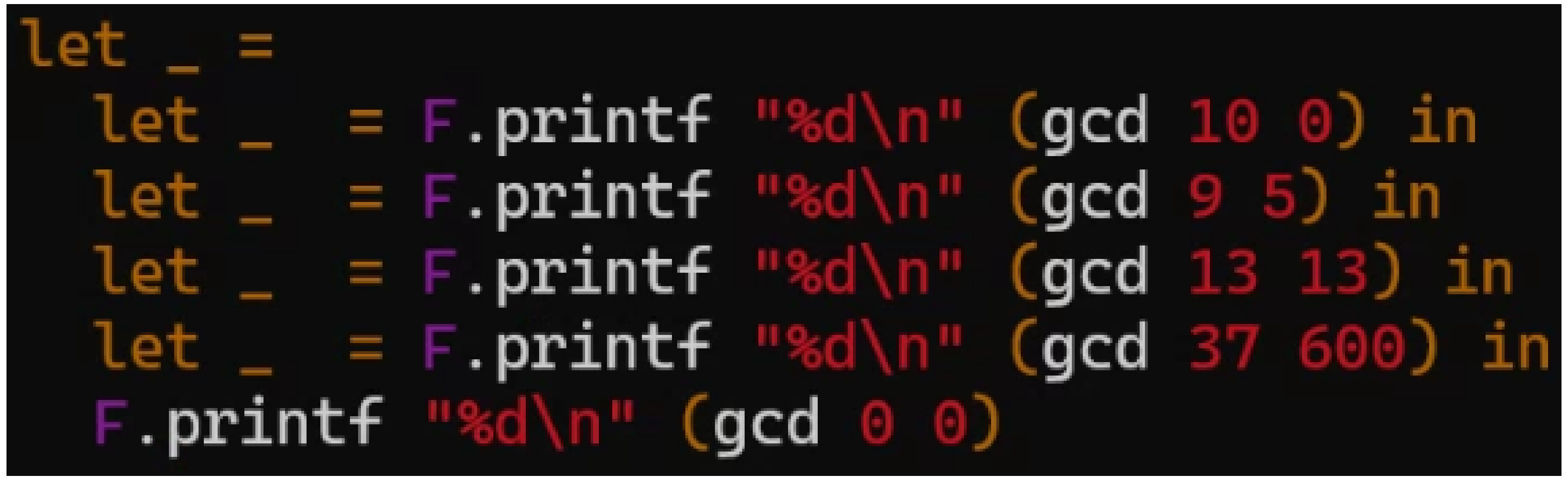
- Don't miss the below instruction.

  1. 7 source code files should be named properly.
  2. Double check your output. If output is not same, you will get deduction.
  3. When compressing your file, there sould be only 7 source code files.
  - No additional folders, or project/dummy files.
  4. Double check your source code is compiled properly.

Frequent Q&A

Q) How we give inputs to the program?

A) Write the input in the .ml file. Like the screenshot below.



```
let _ =
  let _  = F.printf "%d\n" (gcd 10 0) in
  let _  = F.printf "%d\n" (gcd 9 5) in
  let _  = F.printf "%d\n" (gcd 13 13) in
  let _  = F.printf "%d\n" (gcd 37 600) in
  F.printf "%d\n" (gcd 0 0)
```