

2020 봄학기 프로그래밍언어 프로그래밍 과제 #2 (100점)

지수환
93suhwan@gmail.com

임현승
hsim@kangwon.ac.kr

제출 마감: 4월 27일 월요일 11:59pm

1 개요

- 숙제를 진행하기에 앞서 본 문서를 꼼꼼히 읽어보시길 바랍니다.
- 숙제에 대해서 친구들과 토론하는 것은 괜찮지만 숙제는 반드시 혼자서 작성하시기 바랍니다. 특히 코딩하는 중에 토론을 병행하거나 숙제를 완성한 이후에 다른 학생들에게 도움을 줄 경우 자칫하면 서로 매우 유사한 답안이 도출되어 표절로 판단될 수 있으니 주의하시기 바랍니다. 이 경우 설사 친구에게 소스코드를 보여주지 않았더라도 표절로 판단합니다. 또한 다른 친구의 소스코드를 직접 보고 고쳐주는 경우도 부정행위에 해당합니다. 숙제가 표절로 판단될 경우 정보를 제공한 사람(copyee)과 정보를 도용한 사람(copier) 모두 0점 처리합니다. 여러분이 제출한 숙제는 프로그램 표절 검사기(clone-checker)를 이용해서 표절 여부를 검사합니다.
- 숙제가 어렵거나 이해가 안가는 부분이 있다면 가급적 과목 웹페이지 질의응답 게시판을 활용하세요. TA나 제가 최대한 친절히 도와드리겠습니다. 단, 정답은 가르쳐 드리지 않습니다.

숙제를 진행하기 위해 먼저 과목 웹페이지(<http://eruri.kangwon.ac.kr>) 과제 게시판에서 hw2.zip 파일을 다운받아 압축을 풉니다:

```
hsim@ubuntu:~/tmp$ unzip hw2.zip
Archive:  hw2.zip
  creating: hw2/
  inflating: hw2/.depend
  inflating: hw2/hw2.mli
  inflating: hw2/hw2.ml
  inflating: hw2/Makefile
```

반드시 hw2.ml 파일만 수정하세요. hw2.ml 파일은 다음과 같습니다:

```
let sum n = 0;;
let circle r = 0.0;;
let concat s = "";;
let xor x y = true;;
let triangle x y z = true;;
let int_if_then_else b x y = 0;;
let sum_of_fun_val a b c d e = 0;;
let comp3 a b c d = 0;;
let string2 s = "";;
let string256 s = "";;
```

1번 문제인 함수 sum을 작성하기 위해, 등호의 오른쪽에 있는 함수의 몸통인 0를 삭제하고 문제가 요구하는 올바른 연산식을 채워 넣습니다. 나머지 문제들도 동일한 방식으로 작성하면 됩니다.

2 문제

Question 1. [10점] 1에서 n 까지의 정수의 합을 구하는 함수 `sum`을 작성하라.

(타입) `int -> int`

(설명) `sum n`은 $n \leq 0$ 인 경우 0을, $n > 0$ 인 경우 $\frac{n(n+1)}{2}$ 을 반환한다.

(실행 예)

```
# sum;;
- : int -> int = <fun>
# sum (-10);;
- : int = 0
# sum 100;;
- : int = 5050
```

Question 2. [10점] 반지름이 r 인 원의 넓이를 구하는 함수 `circle`을 작성하라.

(타입) `float -> float`

(설명) `circle r`은 $r \leq 0.0$ 인 경우 0.0을, $r > 0.0$ 인 경우 $3.14r^2$ 을 반환한다.

(실행 예)

```
# circle;;
- : float -> float = <fun>
# circle (-10.1);;
- : float = 0.
# circle 4.2;;
- : float = 55.3896
```

Question 3. [10점] 문자열의 앞에 "Hello "를 삽입하는 함수 `concat`을 작성하라.

(타입) `string -> string`

(설명) `concat s`는 문자열 s 의 앞에 "Hello "를 삽입한 문자열을 반환한다. (Hello 뒤에 공백문자가 있음에 유의하라.)

(실행 예)

```
# concat;;
- : string -> string = <fun>
# concat "Bob!";;
- : string = "Hello Bob!"
# concat "Alice!";;
- : string = "Hello Alice!"
```

Question 4. [10점] 논리연산자 `xor`를 계산하는 함수 `xor`를 작성하라.

(타입) `bool -> bool -> bool`

(설명) `xor x y`는 불린형 값 x 와 y 중 하나만 `true`일 경우에 `true`를, 이외의 경우 `false`를 반환한다.

(실행 예)

```
# xor;;
- : bool -> bool -> bool = <fun>
# xor true true;;
- : bool = false
# xor true false;;
- : bool = true
# xor false true;;
- : bool = true
# xor false false;;
- : bool = false
```

Question 5. [10점] 세 정수를 변의 길이로 가지는 삼각형이 존재하는지 확인하는 함수 `triangle`을 작성하라.

(타입) `int -> int -> int -> bool`

(설명) `triangle x y z`는 x, y, z 중 하나라도 0 또는 음수인 경우 `false`를, x, y, z 가 모두 양수인 경우 x, y, z 를 세 변으로 가지는 삼각형이 존재하면 `true`를, 존재하지 않으면 `false`를 반환한다.

(실행 예)

```
# triangle;;
- : int -> int -> int -> bool = <fun>
# triangle (-3) 3 1;;
- : bool = false
# triangle 3 4 5;;
- : bool = true
# triangle 100 1 2;;
- : bool = false
```

Question 6. [10점] 두 정수의 합과 차 중 하나를 선택하는 함수 `int_if_then_else`를 작성하라.

(타입) `bool -> int -> int -> int`

(설명) `int_if_then_else b x y`는 b 가 `true`이면, $x + y$ 를 `false`이면 $x - y$ 를 반환한다.

(실행 예)

```
# int_if_then_else;;
- : bool -> int -> int -> int = <fun>
# int_if_then_else true 2 100;;
- : int = 102
# int_if_then_else (100 < 2) 2 (-2);;
- : int = 4
```

Question 7. [10점] 임의의 2차 함수에 대하여 두 함수 값의 합을 구하는 함수 `sum_of_fun_val`를 작성하라.

(타입) `int -> int -> int -> int -> int -> int`

(설명) `sum_of_fun_val a b c d e`는 a, b, c 를 각각 2차항, 1차항, 상수항의 계수로 갖는 2차 함수 $f(x) = ax^2 + bx + c$ 에 대하여 $f(d) + f(e)$ 의 값을 반환한다.

(실행 예)

```
# sum_of_fun_val;;
- : int -> int -> int -> int -> int -> int = <fun>
# sum_of_fun_val 1 2 1 3 4;;
- : int = 41
# sum_of_fun_val 1 (-3) (-1) 200 123;;
- : int = 54158
```

Question 8. [10점] 정수를 임의의 2차 함수에 세번 적용한 값을 계산하는 함수 `comp3`를 작성하라.

(타입) `int -> int -> int -> int -> int`

(설명) `comp3 a b c d`는 a, b, c 를 각각 2차항, 1차항, 상수항의 계수로 갖는 2차 함수 $f(x) = ax^2 + bx + c$ 에 대하여 $f(f(f(d)))$ 의 값을 반환한다.

(실행 예)

```
# comp3;;
- : int -> int -> int -> int -> int = <fun>
# comp3 1 1 1 1;;
- : int = 183
# comp3 1 (-2) 1 3;;
- : int = 64
```

Question 9. [10점] 두 번 반복된 문자열을 계산하는 함수 `string2`를 작성하라.

(타입) `string -> string`

(설명) `string2 s`는 문자열 s 가 두 번 반복된 문자열을 반환한다.

(실행 예)

```
# string2;;
- : string -> string = <fun>
# string2 "hi";;
- : string = "hihi"
# string2 "abcde";;
- : string = "abcdeabcde"
```

Question 10. [10점] $2^8 (= 256)$ 번 반복된 문자열을 계산하는 함수 `string256`을 작성하라.

(타입) `string -> string`

(설명) `string256 s`는 문자열 `s`가 $2^8 (= 256)$ 번 반복된 문자열을 반환한다.

(힌트) 9번에서 작성한 함수를 활용한다.

(실행 예)

```
# string256;;
- : string -> string = <fun>
# string256 "a";;
- : string =
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
# string256 "ab";;
- : string =
"ababababababababababababababababababababababababab
ababababababababababababababababababababababababab
bababababababababababababababababababababababababab
ababababababababababababababababababababababababab
bababababababababababababababababababababababababab
ababababababababababababababababababababababababab
bababababababababababababababababababababababababab
ababababababababababababababababababababababababab
bababababababababababababababababababababababababab
ababababababababababababababababababababababababab
bababababababababababababababababababababababababab"
```

3 제출

숙제 작성을 마친 후 반드시 `make`를 실행하여 숙제 프로그램이 컴파일이 잘 되는지 확인하세요. **컴파일이 안 되는 코드를 제출하면 본 숙제는 0점입니다.**

```
hsim@ubuntu:~/tmp/hw2$ ls
Makefile hw2.ml hw2.mli
hsim@ubuntu:~/tmp/hw2$ make
ocamlc -c hw2.mli
ocamlc -c hw2.ml
ocamlc -o hw2 hw2.cmo
```

끝으로 완성한 `hw2.ml` 파일을 과제 게시판에 업로드하면 됩니다.