```
$Id$
Answers to 2017q2-midterm, page 1

________________________________________________________________
Question 1. [2]

- return a value
- throw (raise) an exception
- exit the program
- infinite loop or recursion

________________________________________________________________
Question 2. [2]

let sum = List.fold_left (+) 0;;

let length = List.fold_left (fun x _ -> x + 1) 0;;

________________________________________________________________
Question 3(a). [2]

let rec fold_left fn unit list = match list with
    | [] -> unit
    | x::xs -> fold_left fn (fn unit x) xs;;

________________________________________________________________
Question 3(b). [2]

(define (fold_left fn unit list)
        (if (null? list) unit
            (fold_left fn (fn unit (car list)) (cdr list))))

________________________________________________________________
Question 4. [2]

let rec zipwith fn list1 list2 = match list1, list2 with
    | [], _ -> []
    | _, [] -> []
    | x::xs, y::ys -> fn x y :: zipwith fn xs ys;;

```

Answers to 2017q2-midterm, page 2

Question 5. [2]

```
let rec filter fn list = match list with
    | [] -> []
    | x::xs -> if fn x then x :: filter fn xs
                       else filter fn xs;;
```

Question 6. [1]

```
filter (fun x -> 0 > x) [1;2;-3;-4;8;-3];;
```

Question 7. [3]

(a) highest: unary messages such as: 5 sqrt
(b) middle: binary messages such as: 5 + 6
(c) lowest: keyword messages such as: foo from: 5 to: 10

Question 8. [1]

Fortran: John Backus
Cobol: Grace Hopper
Basic John Kemeny & Thomas Kurtz
Lisp: John McCarthy

Question 9. [1]

```
(define (sum list) (foldl + 0 list))
or:
(define (sum list)
        (define (summ list acc)
                (if (null? list) acc
                    (summ (cdr list) (+ acc (car list)))))
        (summ list 0))
```

Question 10. [2]

Many possible answers, but all should be similar.

Example: use a private array in the implementation of a stack.
When popping an element off the stack, neglecting to set the pointer
in the array to null and let it still point at the object,
then the object will be reachagle, even though it is dead.

Answers to 2017q2-midterm, page 3

Question 11. [3]

```
let max (>?) list = match list with
    | [] -> None
    | x::xs -> let rec max' y ys = match ys with
                   | [] -> y
                   | z::zs -> if y >? z then max' y zs
                                        else max' z zs
               in Some (max' x xs);;
```

Question 12. [2]

```
(define (reverse list)
        (define (rev in out)
                (if (null? in) out
                    (rev (cdr in) (cons (car in) out))))
        (rev list '()))
```

Question 13. [2]

```
(define (map2 op list1 list2)
        (if (or (null? list1) (null? list2)) '()
            (cons (op (car list1) (car list2))
                  (map2 op (cdr list1) (cdr list2))))):w
```

Question 14. [3]

Typo in question: it should ask for add', not mul'.

```
let rec add' list1 list2 carry = match (list1, list2, carry) with
    | list1, [], 0       -> list1
    | [], list2, 0       -> list2
    | list1, [], carry   -> add' list1 [carry] 0
    | [], list2, carry   -> add' [carry] list2 0
    | car1::cdr1, car2::cdr2, carry ->
      let sum = car1 + car2 + carry
      in  sum mod 10 :: add' cdr1 cdr2 (sum / 10);;
```

```

Answers to 2017q2-midterm, page 4

 1.    (B) \beta-reduction

 2.    (C) Alonzo Church

 3.    (D) unreachable

 4.    (D) parametric

 5.    (B) inclusion & parametric

 6.    (B) fold_left

 7.    (B) goto

 8.    (D) 1959 COBOL

 9.    (B) link time

10.    (D) parametric

11.    (D) ||

12.    (D) - : int -> int -> int = <fun>


```