INF3380 Exercise Set 4

for Chapter 4 “Basic Communication Operations”

Exercise 1

Calculate $\log n$, $\lfloor \log n \rfloor$, and $\lceil \log n \rceil$ for the following values of n:

- 3
- 13
- 32
- 123
- 321

Exercise 2

Go through the following algorithm, step by step, while assuming (1) *my_id* is the same as *source*, and (2) *my_id* is different from *source*:

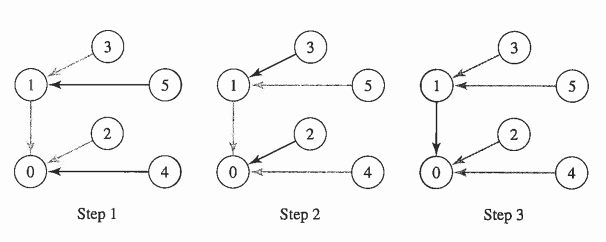
```
procedure GENERAL_ONE_TO_ALL_BC($d$, $my\_id$, $source$, $X$)
begin
    $my\_virtual\_id := my\_id$ XOR $source$;
    $mask := 2^d - 1$;
    for $i := d - 1$ downto 0 do        /* Outer loop */
        $mask := mask$ XOR $2^i$;    /* Set bit $i$ of $mask$ to 0 */
        if ($my\_virtual\_id$ AND $mask$) = 0 then
            if ($my\_virtual\_id$ AND $2^i$) = 0 then
                $virtual\_dest := my\_virtual\_id$ XOR $2^i$;
                send $X$ to ($virtual\_dest$ XOR $source$);
    /* Convert $virtual\_dest$ to the label of the physical destination */
            else
                $virtual\_source := my\_virtual\_id$ XOR $2^i$;
                receive $X$ from ($virtual\_source$ XOR $source$);
    /* Convert $virtual\_source$ to the label of the physical source */
            endelse;
    endfor;
end GENERAL_ONE_TO_ALL_BC
```

Algorithm 4.2 One-to-all broadcast of a message X from *source* on a hypercube.

Exercise 3

Modify the above algorithm so that it can work for any number of processes, not just the power of 2.

Exercise 4



Write a C program that describes the communications performed by an all-to-one reduction operation, shown in the above figure. Given the number of processes as p and a process's ID as i, where $0 \leq i < p$, the C program should print a list of messages sent and/or received by process i. The message list should indicate the destination processor of all sent messages and the source process of all received messages.

For example, for the case where $p = 6$ and $i = 1$, the output of the C program should be

```
Message received from process 5
Message received from process 3
Message sent to process 0
```

Additional exercise (important for oblig 1)

Write a C program that reads in a JPEG image and then

- find the average of its pixel values,
- flip the image in the horizontal direction,
- flip the image in the vertical direction.

The existing C code collection

```
http://heim.ifi.uio.no/xingca/inf-verk3830/simple-jpeg.tar.gz
```

should be used.