1. (35) In the 8-processor butterfly barrier on p.166 of Wilkinson and Allen, suppose that \( P_2 \) goes down after completion of the operation \( P_2 \leftrightarrow P_3 \) in the first stage. Which of the processors, if any, will then wait for the barrier forever?

2. (35) The following code has as its goal edge detection. There are \( N \) rows and \( N \) columns in the image. Fill in the missing code. The variable \( ED \) will control what we might loosely describe as the “amount” of edge detection; the larger \( ED \) is, the sharper the contrast of the edges. For simplicity here, ignore the issue of how we deal with complex numbers.

```c
for (R = 0; R < N; R++)
    for (C = 0; C < N; C++) {
        S = 0.0;
        for (J = 0; J < N; J++)
            for (K = 0; K < N; K++)
                S += x[J][K] * exp(-TwoPiI*(J*R+K*C)/N);
        X[R][C] = S;
    }

    // missing code here
    for (C = 0; C < N; C++) {
        S = 0.0;
        for (J = 0; J < N; J++)
            for (K = 0; K < N; K++)
                S += x[J][K] * exp(TwoPiI*(J*R+K*C)/N);
        x[R][C] = S;
    }
```

3. (30) Write a PerlDSM program which implements the pipelined bubble sort on p.275 of Wilkinson and Allen, in the paragraph which begins with “Bubble sort, as written...” (If you wish, you may write in JIAJIA instead of PerlDSM, but that would be more difficult.) Your program must be of SPMD form, with \( P_k \) handling case \( i = k \) in the outer \( for \) loop in the sequential code shown on that page. Your program must be complete, except that you may omit the code which reads in the array to be sorted from disk. Keep your code short. Make sure to write a draft of your code on scratch paper first, and then copy the clean version to your exam sheet.

Solutions:

1. \( P_0, P_4, P_6 \)