OpenMP 实验内容：

针对以下*题一 ~ 题五*中各个循环，给出完整的 OpenMP C 程序实现，对于不能并行化的，简述原因。

一、 分析以下循环中的依赖关系，并给出相应的迭代依赖图：

```
for i = 2 to 10 do   //循环 1
   for j = 2 to 10
     A[i,j] =  ( A[i-1,j-1] + A[i+1,j+1] )* 0.5;
   endfor
endfor
```

```
for i = 2 to 20 do   // 循环 2
   A[2*i+2] = A[2*i-2] + B[i];
endfor
```

```
for i = 2 to 20 do  // 循环 3
    if A[i] > 0 then
       B[i] = C[i-1] + 1
    else
       C[i] = B[i] – 1
    endif
endfor
```

二、 针对以下两个循环：

```
for i = 1 to M do   //循环 1  M ，N，C 均是常量
   for j = 1 to N
     A[i+1,j+1] =  A[i,j] + C;
   endfor
endfor
```

(1) 给出迭代依赖示意图。

(2) 简述能否逆转外层的 i 循环？能否交换内外循环次序？

```
for i = 1 to 100 do // 循环 2  N 是常量
    X[i] = Y[i] + 10; // 语句 S1
    for j = 1 to 100 do
       B[j] = A[j, N]; // 语句 S2
       for k = 1 to 100 do
          A[j+1, k] = B[j] + C[j, k]; // 语句 S3
       endfor // loop-k
       Y[i+j] = A[j+1, N]; // 语句 S4
    endfor // loop-j
endfor // loop-i
```

(1) 给出此循环的语句依赖图。

(2) 尝试向量化/并行化此循环。

三、 针对以下循环/程序：

```
for i = 1 to 100 do   //循环 1
   for j = 1 to 50 do
      A[3*i+2,2*j-1] =  A[5*j,i+3] + 2;
   endfor
endfor
```

（1） 给出满足依赖方向向量(1,1)的迭代依赖对集合的描述。

（2） 找出与迭代（i=11, j=11）相依赖的迭代（m,n）并指出是哪种依赖？

（3） 能否向量化最内层的 j 循环？如不行，简述理由。

```
S1: x = y * 2
    for i = 1 to 100 do
S2:   C[i] = B[i] + x
S3:   A[i] = C[i-1] + z
S4:   C[i+1] = A[i] * B[i]
       for j = 1 to 50 do
S5:      D[i,j] = D[i,j-1] + x
       endfor
    endfor
S6: z = y + 4
```

给出上述程序的语句依赖图。

四、 分析以下循环中的依赖关系，并给出相应的迭代依赖图：

```
for i = 2 to 10 do   //循环 1
   for j = i to 10
      A[i,j] =  ( A[i,j-1] + A[i-1,j] )* 0.5;
   endfor
endfor
```

```
for i = 1 to 16 do   // 循环 2
   A[i+3] = A[i] + B[i];
endfor
```

```
for k = 1 to 16 step 5 do   // 循环 3 ,k 的循环步长为 5
   for i = k to min(16,i+4) do //设 min 为求最小值函数
      A[i+3] = A[i] + B[i]
   endfor
endfor
```

五、 分析以下 3 个循环中存在的依赖关系；分别通过循环交换、分布和逆转等多种方法来尝试向量化和/或并行化变换：

```
for i = 1 to 100 do   //循环 1
   A[i] =  A[i] + B[i-1];
   B[i] =  C[i-1] * 2 ;
   C[i] =  1 / B[i] ;
   D[i] =  C[i] * C[i] ;
endfor
```

```
for i = 1 to 999 do   // 循环 2
   A[i] = B[i] + C[i];
   D[i] = ( A[i] + A[ 999-i+1 ] ) / 2 ;
endfor
```

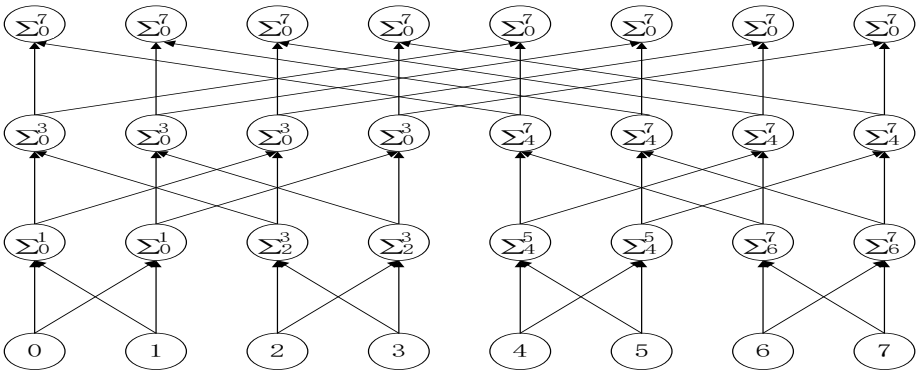
```
for i = 1 to 100 do   // 循环 3
   for j = 1 to 100 do
      A[3*i+2*j, 2*j] = C[i,j] * 2    ;
      D[i,j]          = A[i-j+6, i+j] ;
   endfor
endfor
```

MPI 实验内容

（a） （1.1）写个将 MPI 进程按其所在节点分组的程序；（1.2）在 1.1 的基础上，写个广播程序，主要思想是：按节点分组后，广播的 root 进程将消息“发送”给各组的“0 号”，再由这些“0”号进程在其小组内执行 MPI_Bcast。

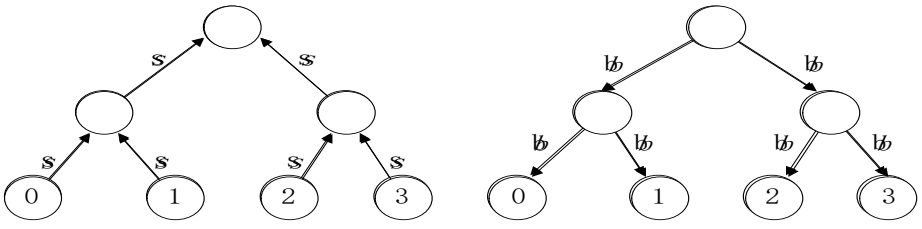
（b） N 个处理器求 N 个数的全和，要求每个处理器均保持全和。

（1）蝶式全和的示意图如下：由于使用了重复计算，共需 logN 步。



给出蝶式全和计算的 MPI 程序实现（设 N 为 2 的幂次方）。

（2）二叉树方式求全和示意图如下：需要 2logN 步。



给出二叉树方式全和计算的 MPI 程序实现。

（c） 《并行算法实践》单元 V 习题 v-3。给出 FOX 矩阵相乘并行算法的 MPI 实现。

（d） 参数服务器系统的 MPI 模拟。

设系统中总计有 N 个进程，其中 P 个进程作为参数*服务器进程*，而 Q 个进程作为*工作进程*（N = P + Q， 且 0 < P < < Q）。工作进程和服务器进程的互动过程如下：

1. 第 i 个工作进程首先产生一个随机数，发送给第 i%P 个参数服务器进程。然后等待并接收它对应的参数服务器进程发送更新后的数值，之后，再产生随机数，再发送……。
2. 每个参数服务器进程等待并接收来自它对应的所有工作进程的数据，在此之后，经通信，使所有的参数服务器获得所有工作进程发送数据的平均值。
3. 每个参数服务器发送该平均值给它对应的所有工作进程，然后再等待……。

试给出上述互动过程的 MPI 程序实现。

（e） 矩阵 A 和 B 均为 N*N 的双精度数矩阵，有 P 个处理器。针对以下程序片段，分别采用按行块连续划分以及棋盘式划分方式，给出相应的 MPI 并行实现。

```
for(i=1; i<N-1; i++)
    for( j=1;j<N-1; j++)
        B[i][j] = (A[i-1][j] + A[i][j+1] + A[i+1][j] + A[i][j-1]) / 4.0
```

个人实验

结合你自己研究方向，设计一个采用并行计算来求解的问题。简要描述你的问题，并从 PCAM 角度来分析你的并行设计方案，给出你的并行实现及加速性能分析与评测（可以选择 MPI 或 OpenMP 或混合并行）。