

1. Explain all dependences and their types that you detect in the following loop:

```
#pragma omp parallel for
for (i=1;i<n;i++) {
    b[i] = a[tab[i]];
    a[tab[i]] = a[i];
}
```

2. Give an example of a simple loop that the Range Test can find parallel but the Banerjee-Wolfe test cannot.

3. Privatization is one of the most important techniques in a parallelizing compiler. Give an intuitive explanation, why this is so.
4. Write an example loop that can be found parallel (by any DD test you know) if and only if the techniques array privatization, parallel reduction recognition, and induction variable substitution have been applied. Also write the parallel version of the loop. Explain, briefly.

5. Flow dependences are also called *true dependences*, Anti and output dependences are called *memory-related dependences*. Explain why.

6. Imagine you are an optimizing compiler algorithm that has to decide which one of the following loop versions is better. Explain your decision making process.

```
DO i=1,n
  DO j=1,m
    A(i,j) = A(i,j)+A(i,j-
1)
  ENDDO
ENDDO
```

```
DO j=1,m
  DO i=1,n
    A(i,j) = A(i,j)+A(i,j-
1)
  ENDDO
ENDDO
```

7. Bonus: Write an interesting question for the ECE663 final exam, and supply its answer.