An *associative operation* can be regrouped in any way while producing the same result. Integer arithmetic and integer modulo arithmetic (as we have in C) are associative.

As an example, the following all compute the same value:

- (((((((a[0]+a[1])+a[2])+a[3])+a[4])+a[5])+a[6])+a[7])
- (a[0]+(a[1]+(a[2]+(a[3]+(a[4]+(a[5]+(a[6]+a[7])))))))
- (((a[0]+a[1])+(a[2]+a[3]))+((a[4]+a[5])+(a[6]+a[7])))

1. Consider the following sum operation:

```
int sum=0;
for (int i=0;i<N; i++)
    sum+=a[i];
```

   (a) Assuming addition takes one cycle, what is the II for the computation?

   (b) What is the latency bound?

   (c) Is this computation data parallel?

2. What other common operations are associative?

3. Consider the following prefix-sum operation:

```
int sum[N];
sum[0]=a[0];
for (int i=1;i<N; i++)
    sum[i]=a[i]+sum[i-1];
```

   (a) What is the latency bound?

   (b) How much hardware (area) does it require to achieve within a factor of two of this latency bound? (assume unit area adders)

4. Is Majority(A,B,C) = (A &&B) || (A && C) || (B && C) associative?
   Consider:

   - Majority(1,1,Majority(1,1,Majority(1,0,0)))
   - Majority(Majority(Majority(1,1,1),1,1),0,0)