

1. O bubble sort ocorre por flutuação, sendo que em cada repetição, exige um vetor em que seus elementos são organizados do maior até o menor na sequência. Logo, por ser um algoritmo de ordenação (parallel e o bubble sort) é necessário que existam dados, caso o contrário, a não existência de dados torna a comparação inexistente. ,
2. Pode ser paralelizado utilizando apenas usando o #pragma open parallel nos dois comandos for.

```
37 void evenOddSort(int A[], int n){
38     #pragma open parallel
        for(int i = 0; i<n; i++){
39         int first = i%2;
40         #pragma open parallel
            for(int j = first; j<n-1; j += 2){
41             if(A[j]>A[j+1]){
42                 swap(A[j],A[j+1]);
43             }
44         }
45     }
46 }
```

3.

```
37 #include<time.h>
38 #include<stdio.h>
39 #include<stdlib.h>
40 #include<omp.h>
41 int main(int argc, char **argv){
42     int n = atoi(argv[1]);
43     int a[n];
44     int even[n];
45     int count = 0;
46     srand(time(NULL));
47     #pragma open parallel
        for(int i = 0; i<n; i++){
48         a[i] = rand()%n;
49     }
50     #pragma open parallel
        for(int i = 0; i<n; i++){
51         even[i] = (a[i]%2==0)?1:0;
52     }
53     #pragma open parallel
        for(int i = 0; i<n; i++){
54         count += even[i];
55     }
56     printf("even numbers: %d\n", count);
57     return 0;
58 }
59 }
```

```

4. 37 #include <stdio.h>
    38 #include<math.h>
    39 int is_prime(long num){
    40     if(num<=1)
    41         return 0;
    42     else if(num>3){
    43         if(num%2==0)
    44             return 0;
    45         long max_divisor = sqrt(num);
    46         #pragma open parallel

        for(long d = 3; d<=max_divisor; d+=2){
    47             if(num%d==0)
    48                 return 0;
    49         }
    50     }
    51     return 1;
    52 }

37 int main(){
38     long max_num = 500000L;
39     long sum;
40
41     #pragma open parallel
42     if(max_num<=1) sum = 0;
43     #pragma open parallel
44     else if(max_num==2) sum = 1;
45     else{
46         sum = 1; //count the 2 as a prime, then start
from 3
47
48         #pragma open parallel
49         for(int n = 3; n<=max_num; n += 2){ //skip
all even numbers
50             long count_prime = is_prime(n);
51             sum = sum + count_prime;
52         }
53     }
54     printf("maximum number checked: %ld\n",
max_num);
55     printf("number of primes: %ld\n", sum);
56     return 0;
57 }

```