```
// Algorithms 1 and 4 from Weiss, DSAAC++
textbook;
// computing of maximal subsum of a vector of
positive and
// negative values;

#include <iostream>
#include <vector>
#include "MaxSubSum.h"

int max_sub_sum_alg1(const vector<int>& vec,
int& ops)
{
    int maxsum = 0;
    ops = 0;

    for (int i = 0; i < vec.size(); i++)
    {
        for (int j = 0; j < vec.size(); j++)
        {
            int localsum = 0;
            for (int k = i; k <= j; k++)
            {
                localsum += vec[k];
                ops++;
            }

            if (localsum > maxsum)
                maxsum = localsum;
        }
    }
    return maxsum;
}
```

```
int max_sub_sum_alg4(const vector<int>& vec,
int& ops)
{
    int maxsum = 0;
    int localsum = 0;
    ops = 0;

    for (int i = 0; i < vec.size(); i++)
    {
        localsum += vec[i];
        ops++;

        if (localsum > maxsum)
            maxsum = localsum;
        else if (localsum < 0)
            localsum = 0;
    }
    return maxsum;
}
```