

Problem 6: The Subvector Problem¹

Source filename: `subvector.(cpp|java)`
 Input filename: `subvector.in`
 Output filename: `subvector.out`

Design an algorithm to read a vector (an array) of n integers ($1 \leq n \leq 1,000,000$) and then finds the maximum sum found in any *contiguous* subvector of the array. The first line in the input file, `subvector.in`, contains an integer m ($0 < m \leq 10$) which determines the number of vectors stored in the file that should be read. The next line will contain at least 1 integer. This first integer (there may be more) on the second line will correspond to n , the number of elements in the 1st vector. The next n integers (they may be stored on multiple lines, but all are separated from all other integers by one or more spaces or end-of-line characters) correspond to the n elements of the vector. Every value for each of the n elements is bounded by $-10000 \leq \text{value} \leq 10000$; however, at least one of the elements will be nonnegative.

When your program finds the maximum sum found in any contiguous subvector of the given vector, it should send one line of output to the output file, `subvector.out`. This line should simply report the maximum sum.

Once your program has found the maximum sum for one vector it should read a new vector. The next unread integer will be the new value for n ; and this will be followed by the n values for the n elements of the next vector. Find the maximum sum for this vector and report the result on a new line in the output file.

After m vectors have been read and their maximum sums found, your program should quit.

Sample Input File (`subvector.in`)

```
2
10 31 -41 59      26 -53 58    97 -93 -23 84
5 13
-20
8
-8 9
```

Sample Output File (`subvector.out`)

```
187
13
```

¹ The student will be well served by reading the excellent book by Jon Bentley entitled, Programming Pearls.