**Max Sum Plus Plus**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 65536/32768 K (Java/Others)  
Total Submission(s): 18114    Accepted Submission(s): 5926**

**Problem Description**

Now I think you have got an AC in Ignatius.L's "Max Sum" problem. To be a brave ACMer, we always challenge ourselves to more difficult problems. Now you are faced with a more difficult problem.  
  
Given a consecutive number sequence S*1*, S*2*, S*3*, S*4* ... S*x*, ... S*n* (1 ≤ x ≤ n ≤ 1,000,000, -32768 ≤ S*x* ≤ 32767). We define a function sum(i, j) = S*i* + ... + S*j* (1 ≤ i ≤ j ≤ n).  
  
Now given an integer m (m > 0), your task is to find m pairs of i and j which make sum(i*1*, j*1*) + sum(i*2*, j*2*) + sum(i*3*, j*3*) + ... + sum(i*m*, j*m*) maximal (i*x* ≤ i*y* ≤ j*x* or i*x* ≤ j*y* ≤ j*x* is not allowed).  
  
But I`m lazy, I don't want to write a special-judge module, so you don't have to output m pairs of i and j, just output the maximal summation of sum(i*x*, j*x*)(1 ≤ x ≤ m) instead. ^\_^

**Input**

Each test case will begin with two integers m and n, followed by n integers S*1*, S*2*, S*3* ... S*n*.  
Process to the end of file.

**Output**

Output the maximal summation described above in one line.

**Sample Input**

1 3 1 2 3

2 6 -1 4 -2 3 -2 3

**Sample Output**

6

8

***Hint***

Huge input, scanf and dynamic programming is recommended.

**Author**

JGShining（极光炫影）

**Recommend**

We have carefully selected several similar problems for you:  [1074](http://acm.hdu.edu.cn/showproblem.php?pid=1074) [1081](http://acm.hdu.edu.cn/showproblem.php?pid=1081) [1080](http://acm.hdu.edu.cn/showproblem.php?pid=1080) [1160](http://acm.hdu.edu.cn/showproblem.php?pid=1160) [1114](http://acm.hdu.edu.cn/showproblem.php?pid=1114)