Submissions

Testrun

0pt	Not attempted **0/142 users** correct (0%)

sandwich

1pt	Not attempted **187/205 users** correct (91%)
15pt	Not attempted **141/178 users** correct (79%)

majority

1pt	Not attempted **170/176 users** correct (97%)
20pt	Not attempted **80/167 users** correct (48%)

shhhh

1pt	Not attempted **110/115 users** correct (96%)
30pt	Not attempted **69/102 users** correct (68%)

load_balance

2pt	Not attempted **94/101 users** correct (93%)
35pt	Not attempted **33/88 users** correct (38%)

Top Scores

iwi	105
simonlindholm	105
Murphy	105
stgatilov	105
Alexander86	105
microtony	105
eatmore	105
uwi	105
Marcin.Smulewicz	105
tczajka	105

Problem B. sandwich

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

small 1 points *2 minute timeout*	The contest is finished.
large 15 points *10 minute timeout*	The contest is finished.

Problem

Your friends made a brave attempt to beat the world record for making the longest sub sandwich ever. They failed or succeeded (it doesn't really matter), and proposed to you to eat as much of the sandwich as you want to.
The sandwich is composed of N parts. Each part has a taste value for you, which might be positive (meaning you want to eat it) or negative (meaning you would prefer not to). Ideally, you would just eat the tasty parts, but it's rude to break out the middle of the sandwich. So, instead you want to eat some part from the beginning, and some part from the end; and to make the total taste of what you eat as large as possible. The total taste is the sum of tastes of all the parts you have eaten.
Note: it's OK to eat the whole sandwich, or to eat nothing at all. Output the largest total taste of what you can eat.

Input

The input library will be called "sandwich", see the sample inputs below for examples in your language. It will define two methods: GetN(), which will return the number of parts of the sandwich, and GetTaste(i), which will return the taste of the *i*th part of the sandwich, for $0 \le i < N$.
A single call to GetTaste(i) will take approximately 0.01 microseconds.

Output

Output one number: the maximum possible total taste of the parts you will eat.

Limits

Each node will have access to 128MB of RAM, and a time limit of 3 seconds.
$-10^9 \le$ GetTaste(i) $\le 10^9$ for all i with $0 \le i <$ GetN().

Small input

Your solution will run on 10 nodes.
$1 \le$ GetN() ≤ 1000.

Large input

Your solution will run on 100 nodes.
$1 \le$ GetN() $\le 5 \times 10^8$.

Sample

```
Input                          Output

See the input files below.     For sample input 1:
                               14
                               For sample input 2:
                               0
                               For sample input 3:
                               5
```

Note: this problem idea was used by us in the practice round of the Algorithmic Engagements contest in 2014, and also displayed as an example in our FAQ.

Sample input libraries:
Sample input for test 1: sandwich.h [CPP] sandwich.java [Java]
Sample input for test 2: sandwich.h [CPP] sandwich.java [Java]
Sample input for test 3: sandwich.h [CPP] sandwich.java [Java]