Dessert Wizard

Problem code: DELISH

All submissions for this problem are available.

It's finally summer in Chefland! So our chef is looking forward to prepare some of the best "beat-the-heat" dishes to attract more customers. He summons the Wizard of Dessert to help him with one such dish.

The wizard provides the chef with a sequence of **N** ingredients where the i^{th} ingredient has a delish value of **D[i]**. The preparation of the dish takes place in two phases.

Phase 1 : The chef chooses two indices **i** and **j** and adds the ingredients **i, i+1, ..., j** to his dish. He also finds the sum of the delish value in this range i.e **D[i] + D[i+1] + ... + D[j]**.

Phase 2 : The chef chooses two more indices **k** and **l** and adds the ingredients **k, k+1, ..., l** to his dish. He also finds the sum of the delish value in this range i.e **D[k] + D[k+1] + ... + D[l]**.

Note that **1 ≤ i ≤ j < k ≤ l ≤ N**.

The total delish value of the dish is determined by the absolute difference between the values obtained in the two phases. Obviously, the chef wants to maximize the total delish value for his dish. So, he hires you to help him.

Input

First line of input contains an integer **T** denoting the number of test cases. For each test case, the first line contains an integer **N** denoting the number of ingredients. The next line contains **N** space separated integers where the i^{th} integer represents the delish value **D[i]** of the i^{th} ingredient.

Output

Print the maximum delish value of the dish that the chef can get.

Constraints

- **1 ≤ T ≤ 50**
- **2 ≤ N ≤ 10000**
- **-1000000000 (-10^9) ≤ D[i] ≤ 1000000000 (10^9)**

Example

Input:
2
5
1 2 3 4 5
4
1 1 -1 -1

Output:
13
4

Explanation

Example case 1.

SUCCESSFUL SUBMISSIONS

User	Time	Mem	Lang	Solution
shacer28	0.07	22.4M	C++ 4.9.2	View
deepakgupta13	0.09	22.4M	C++ 4.9.2	View
jeet343	0.10	21M	C	View
shashwat_01	0.16	2.9M	C++ 4.0.0-8	View
rodomonte	0.17	1.8M	C	View
ry_7	0.17	2.8M	C++ 4.3.2	View
psych_cod3r	0.18	2.2M	C	View
betrayermor	0.18	3.5M	C++ 4.8.1	View
abhiabhishek	0.20	1.9M	C	View
parvez_12121	0.20	3M	C++ 4.3.2	View
aravind159	0.20	3M	C++ 4.3.2	View
rrav	0.22	2.5M	C	View

1 of 25 Next »

HELP

Program should read from **standard input** and write to **standard output**. After you submit a solution you can see your results by clicking on the **[My Submissions]** tab on the problem page. Below are the possible results:

- **Accepted** Your program ran successfully and gave a correct answer. If there is a score for the problem, this will be displayed in parenthesis next to the checkmark.
- **Time Limit Exceeded** Your program was compiled successfully, but it didn't stop before time limit. Try optimizing your approach.
- **Wrong Answer** Your program compiled and ran succesfully but the output did not match the expected output.

Chef can choose **i = j = 1, k = 2, l = 5**.

The delish value hence obtained is **| (2+3+4+5) − (1) | = 13** .

Example case 2.

Chef can choose **i = 1, j = 2, k = 3, l = 4**.

The delish value hence obtained is **| ((−1) + (−1)) − (1 + 1) | = 4** .

Author:	viv001
Tester:	tuananh93
Editorial	http://discuss.codechef.com/problems/DELISH
Tags	dynamic-prog june13 simple viv001
Date Added:	30-03-2013
Time Limit:	1 sec
Source Limit:	50000 Bytes
Languages:	ADA, ASM, BASH, BF, C, C99 strict, CAML, CLOJ, CLPS, CPP 4.3.2, CPP 4.9.2, CPP14, CS2, D, ERL, FORT, FS, GO, HASK, ICK, ICON, JAVA, JS, LISP clisp, LISP sbcl, LUA, NEM, NICE, NODEJS, PAS fpc, PAS gpc, PERL, PERL6, PHP, PIKE, PRLG, PYTH, PYTH 3.1.2, RUBY, SCALA, SCM guile, SCM qobi, ST, TCL, TEXT, WSPC

SUBMIT

- **Runtime Error** Your code compiled and ran but encountered an error. The most common reasons are using too much memory or dividing by zero. For the specific error codes see the help section.
- **Compilation Error** Your code was unable to compile. When you see this icon, click on it for more information.

If you are still having problems, see a sample solution here.

Comments

daddycoder *@ 17 Jun 2013 03:50 PM*

I have used the same approach as many other ACs(the ones in Java too). Why TLE? Submission : http://www.codechef.com/viewsolution/2276466

sumanth232 *@ 17 Jun 2013 05:17 PM*

Let j....i be the indices , where u get the max sum . let sum(i) = a[0] + a[1] + ... + a[i]. The sum = sum(i) - sum(j-1) .Your aim is to maximize sum(i) and minimize sum(j-1) . So, in one traversal u can just find the maximum sum(i) and minimum sum(j-1) , ofcourse with the conditions 0 <= j <= i <= n-1 . through this approach, u dont need to worry about the limitations of Kadanes algorithm

daddycoder *@ 17 Jun 2013 11:03 PM*

Why won't we consider minimizing sum(i) and maximizing sum(j-1), that would also yield certain sum that might be greater than other sums. And that requires another traversal. Is the approach what you suggest exhaustive for this problem. How? The limitations of kadane(of negative numbers) is already taken into account.

daddycoder *@ 17 Jun 2013 11:06 PM*

Here's a solution that got a green mark which follows the same idea. http://www.codechef.com/viewsolution/2233152

fu_k9 *@ 18 Jun 2013 03:01 PM*

http://www.codechef.com/viewsolution/2278153 i have tested my code against 50 test cases with delish value of order 10^9 and 10^-9..i got 12 millisecond running time..but m still getting TLE..please help ..

rehpotsirhc *@ 16 Feb 2015 07:12 AM*

Don't use a regular int... use long

Need help? Post a comment. But before that please spare a moment to read the guidelines.
Your name:
akashiitj

Comment: *

Save

CodeChef is a non-commercial competitive programming community

CodeChef - A Platform for Aspiring Programmers

CodeChef was created as a platform to help programmers make it big in the world of algorithms, **computer programming** and **programming contests**. At CodeChef we work hard to revive the geek in you by hosting a **programming contest** at the start of the month and another smaller programming challenge in the middle of the month. We also aim to have training sessions and discussions related to **algorithms, binary search**, technicalities like **array size** and the likes. Apart from providing a platform for **programming competitions**, CodeChef also has various algorithm tutorials and forum discussions to help those who are new to the world of **computer programming**.

Practice Section - A Place to hone your 'Computer Programming Skills'

Try your hand at one of our many practice problems and submit your solution in a language of your choice. Our **programming contest** judge accepts solutions in over 35+ programming languages. Preparing for coding contests were never this much fun! Receive points, and move up through the CodeChef ranks. Use our practice section to better prepare yourself for the multiple **programming challenges** that take place through-out the month on CodeChef.

Compete - Monthly Programming Contests and Cook-offs

Here is where you can show off your **computer programming** skills. Take part in our 10 day long monthly **coding contest** and the shorter format Cook-off **coding contest**. Put yourself up for recognition and win great prizes. Our **programming contests** have prizes worth up to INR 20,000 (for Indian Community), $700 (for Global Community) and lots more CodeChef goodies up for grabs.