Profits: 113.00 Rank: 64						Pades Deserve	(Detrille)		
Problem   Submissions   Leaderboard   Discussions	Dashboard > Functional Programming > Introduction > Update List						Points: 113.00 Rank: 6454		
Lipidate the values of a list with their absolute values. The input and output portions will be handled automatically during grading. You only need to write a function with the recommended method signature.  Input Format  Dutput Format  Dutput He absolute value of N integers, each on a separate line. These are the N elements of the input array.  Dutput He absolute value of N integers, each on a separate line in the same input order.  Sample Input  2  1  1  1  2  3  3  4  3  5  Sample Output  2  4  3  5  Sample Output  2  6  8  Constraints  The list will have no more than 100 integers. Each integer, X, in the list −100 < − X < − 100.  Recommended Method Signature  Number of Parameters: 1  Parameters: 1    Tarameters: 1    Parameters: 1   Tarameters: 1    Parameters: 1   Tarameters: 1    Parameters: 1   Tarameters: 1    Parameters: 1    Paramete									
with the recommended method signature.  **pout Format**  **Dutput Format**  **Dutput Format**  **Dutput How absolute value of **N* integers, each on a separate line in the same input order.  **Sample Input**  **Dutput How absolute value of **N* integers, each on a separate line in the same input order.  **Sample Input**  **Parameters**  **Parameters**  **Parameters**  **In the list — 100 <= **X <= 100.**  **Recommended Method Signature**  **Number**  **Number**  **Number**  **Number**  **Number**  **Of - Parameters**: 11   11   11   11   11   11   11   1	Problem	Submissions	Leaderboard	Discussions					
There are $N$ integers, each on a separate line. These are the $N$ elements of the input array.  Dutput Format  Dutput the absolute value of $N$ integers, each on a separate line in the same input order.  Sample Input  2 4 3 3 -1 23 -4 -54 -54 -54 -54 -54 -54 -55  Constraints  The list will have no more than 100 integers.  Sach integer, $X$ , in the list: $-100 <= X <= 100$ .  Recommended Method Signature  Musher of Parameters: 1 Para	-		te values. The input and	d output portions will be ha	ındled automati	cally during gradi	ng. You o	nly need to	write a function
Dutput the absolute value of <i>N</i> integers, each on a separate line in the same input order.  Sample Input  2  4 3 3 1-2 3 -4 -54 -54 -54 -54 -54 -54 -54 -54 -54	nput Format								
Couptut the absolute value of <b>N</b> integers, each on a separate line in the same input order.  Sample Input  2	There are $m{N}$ into	egers, each on a separate l	ne. These are the $m{N}$ ele	ements of the input array.					
Sample Input  2 4 3 3 -1 23 -54  Sample Output  2 4 3 3 3 1 23 4 5 5  Constraints  The list will have no more than 100 integers. Each integer, X, in the list: -100 <= X <= 100.  Recommended Method Signature  Number of Parameters: 1 Parameters: (1sts) Returns: List or Vector  For Hackers Using Clojure  This will be the outline of your function body (fill in the blank portion marked by underscores):  (fn[tst]	Output Format								
2 3 3 3 3 3 3 3 3 3 3 3 3 4 -54  Sample Output  2 4 3 3 3 5 5  Constraints  The list will have no more than 100 integers. Each integer, X, in the list: -100 <= X <= 100.  Recommended Method Signature  Number of Parameters: 1 Parameters: 11st1 Par	Output the abso	olute value of <b>N</b> integers, e	ach on a separate line i	n the same input order.					
-4 -54 -54 -54 -54 -54 -54 -54 -54 -54 -	Sample Input								
2 4 3 1 1 23 4 54 54 55  The list will have no more than 100 integers. Each integer, <i>X</i> , in the list: $-100 <= X <= 100$ .  Recommended Method Signature  Number 0f Parameters: 1 Parameters: [List] Returns: List or Vector  For Hackers Using Clojure  This will be the outline of your function body (fill in the blank portion marked by underscores):  (fn[lst])  For Hackers Using Scala  This will be the outline of your function body (fill in the blank portion marked by underscores):  def f(arr:List[Int]):List[Int] =  For Hackers Using Haskell  This will be the outline of your function body (fill in the blank portion marked by underscores):	-4 3 -1 23 -4								
4 3 1 23 4 54  Constraints  The list will have no more than 100 integers. Each integer, <i>X</i> , in the list: -100 <= <i>X</i> <= 100.  Recommended Method Signature  Number of Parameters: 1 Parameters: [list] Returns: List or Vector  For Hackers Using Clojure  This will be the outline of your function body (fill in the blank portion marked by underscores):  (fn[lst]	Sample Output								
The list will have no more than 100 integers. Each integer, X, in the list: -100 <= X <= 100.  Recommended Method Signature  Number Of Parameters: 1 Parameters: [list] Returns: List or Vector  For Hackers Using Clojure  This will be the outline of your function body (fill in the blank portion marked by underscores):  (fn[lst])  For Hackers Using Scala  This will be the outline of your function body (fill in the blank portion marked by underscores):  def f(arr:List[Int]):List[Int] =  For Hackers Using Haskell  This will be the outline of your function body (fill in the blank portion marked by underscores):	4 3 1 23 4								
Each integer, <i>X</i> , in the list: $-100 <= X <= 100$ .  Recommended Method Signature  Number Of Parameters: 1 Parameters: [list] Returns: List or Vector  For Hackers Using Clojure  This will be the outline of your function body (fill in the blank portion marked by underscores):  (fn[lst])  For Hackers Using Scala  This will be the outline of your function body (fill in the blank portion marked by underscores):  def f(arr:List[Int]):List[Int] =  For Hackers Using Haskell  This will be the outline of your function body (fill in the blank portion marked by underscores):									
Number Of Parameters: 1 Parameters: [List] Returns: List or Vector  For Hackers Using Clojure  This will be the outline of your function body (fill in the blank portion marked by underscores):  (fn[lst])  For Hackers Using Scala  This will be the outline of your function body (fill in the blank portion marked by underscores):  def f(arr:List[Int]):List[Int] =  For Hackers Using Haskell  This will be the outline of your function body (fill in the blank portion marked by underscores):									
Parameters: [list] Returns: List or Vector  For Hackers Using Clojure  This will be the outline of your function body (fill in the blank portion marked by underscores):  (fn[lst])  For Hackers Using Scala  This will be the outline of your function body (fill in the blank portion marked by underscores):  def f(arr:List[Int]):List[Int] =  For Hackers Using Haskell  This will be the outline of your function body (fill in the blank portion marked by underscores):	Recommended	Method Signature							
This will be the outline of your function body (fill in the blank portion marked by underscores):  (fn[lst])  For Hackers Using Scala  This will be the outline of your function body (fill in the blank portion marked by underscores):  def f(arr:List[Int]):List[Int] =  For Hackers Using Haskell  This will be the outline of your function body (fill in the blank portion marked by underscores):	Parameters:	[list]							
(fn[lst])  For Hackers Using Scala  This will be the outline of your function body (fill in the blank portion marked by underscores):  def f(arr:List[Int]):List[Int] =  For Hackers Using Haskell  This will be the outline of your function body (fill in the blank portion marked by underscores):	For Hackers Us	ing Clojure							
For Hackers Using Scala  This will be the outline of your function body (fill in the blank portion marked by underscores):  def f(arr:List[Int]):List[Int] =  For Hackers Using Haskell  This will be the outline of your function body (fill in the blank portion marked by underscores):			dy (fill in the blank portion	on marked by underscores)	:				
This will be the outline of your function body (fill in the blank portion marked by underscores):  def f(arr:List[Int]):List[Int] =  For Hackers Using Haskell  This will be the outline of your function body (fill in the blank portion marked by underscores):	(fn[lst]_		_)						
def f(arr:List[Int]):List[Int] =  For Hackers Using Haskell  This will be the outline of your function body (fill in the blank portion marked by underscores):	For Hackers Us	ing Scala							
For Hackers Using Haskell  This will be the outline of your function body (fill in the blank portion marked by underscores):			dy (fill in the blank portion	on marked by underscores)	:				
This will be the outline of your function body (fill in the blank portion marked by underscores):	def f(arr:	List[Int]):List[Int] =							
This will be the outline of your function body (fill in the blank portion marked by underscores):	For Hackers Us	ing Haskell							
			dy (fill in the blank portion	on marked by underscores)	:				
f arr =	£								

1 of 2 4/19/17, 18:11 PM

You have to read input from standard input and write output to standard output. Please follow the input/output format mentioned above.

NOTE: You only need to submit the code above after filling in the blanks appropriately. The input and output section will be handled by us. The focus is on writing the correct function.



Current Buffer (saved locally, editable)  $\ \mathcal{V} \ \mathfrak{O}$ Racket 1 #lang racket 2 3 (define (f lst) (map abs 1st) 5 7 (define (read-list) 8 (let ([x (read)])
 (if (eof-object? x) 10 (list) 11 (cons x (read-list))))) 12 (let ([lst (read-list)]) 13 (let ([ans (f lst)]) 14 (for ([x ans]) (printf "~a\n" x)))) 15 16 17 Line: 1 Col: 1 **1** Upload Code as File Run Code Test against custom input Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Copyright © 2017 HackerRank. All Rights Reserved

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature

2 of 2 4/19/17, 18:11 PM