

Status	Finished
Started	Sunday, 20 October 2024, 1:26 PM
Completed	Monday, 21 October 2024, 7:45 PM
Duration	1 day 6 hours
Marks	8.00/8.00
Grade	10.00 out of 10.00 (100%)



Question 1

Correct

Mark 1.00 out of 1.00

Implement all methods in class **Stack** with template type **T**. The description of each method is written as comment in frame code.

```
#ifndef STACK_H
#define STACK_H
#include "DLinkedList.h"
template<class T>
class Stack {
protected:
    DLinkedList<T> list;
public:
    Stack() {}
    void push(T item) ;
    T pop() ;
    T top() ;
    bool empty() ;
    int size() ;
    void clear() ;
};

#endif
```

You can use all methods in class **DLinkedList** without implementing them again. The description of class **DLinkedList** is written as comment in frame code.

```
template <class T>
class DLinkedList
{
public:
    class Node;    //forward declaration
protected:
    Node* head;
    Node* tail;
    int count;
public:
    DLinkedList() ;
    ~DLinkedList();
    void add(const T& e);
    void add(int index, const T& e);
    T removeAt(int index);
    bool removeItem(const T& removeItem);
    bool empty();
    int size();
    void clear();
    T get(int index);
    void set(int index, const T& e);
    int indexOf(const T& item);
    bool contains(const T& item);
};
```

For example:

Test	Result
Stack<int> stack; cout << stack.empty() << " " << stack.size();	1 0
Stack<int> stack; int item[] = { 3, 1, 4, 5, 2, 8, 10, 12 }; for (int idx = 0; idx < 8; idx++) stack.push(item[idx]); assert(stack.top() == 12); stack.pop(); stack.pop(); cout << stack.top();	8

Answer: (penalty regime: 0 %)

Reset answer

```

1 void push(T item) {
2     // TODO: Push new element into the top of the stack
3     list.add(0,item);
4     //cout<<list.get(0);
5 }
6
7 T pop() {
8     // TODO: Remove an element on top of the stack
9     T ret=list.get(0);
10    list.removeAt(0);
11    return ret;
12 }
13
14 T top() {
15     // TODO: Get value of the element on top of the stack
16     //cout<<list.get(0);
17     return list.get(0);
18 }
19
20 bool empty() {
21     // TODO: Determine if the stack is empty
22     return list.empty();
23 }
24
25 int size() {
26     // TODO: Get the size of the stack
27     return list.size();
28 }
29
30 void clear() {
31     // TODO: Clear all elements of the stack
32     list.clear();
33 }

```

	Test	Expected	Got	
✓	<pre>Stack<int> stack; cout << stack.empty() << " " << stack.size();</pre>	1 0	1 0	✓
✓	<pre>Stack<int> stack; int item[] = { 3, 1, 4, 5, 2, 8, 10, 12 }; for (int idx = 0; idx < 8; idx++) stack.push(item[idx]); assert(stack.top() == 12); stack.pop(); stack.pop(); cout << stack.top();</pre>	8	8	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 1.00 out of 1.00

You are keeping score for a basketball game with some new rules. The game consists of several rounds, where the scores of past rounds may affect future rounds' scores.

At the beginning of the game, you start with an empty record. You are given a list of strings **ops**, where **ops[i]** is the operation you must apply to the record, with the following rules:

- A non-negative integer **x (from 0 to 9)** - record a new score of **x**
- **'+'** - Record a new score that is the sum of the previous two scores. It is guaranteed there will always be two previous scores.
- **'D'** - Record a new score that is double the previous score. It is guaranteed there will always be a previous score.
- **'C'** - Invalidate the previous score, removing it from the record. It is guaranteed there will always be a previous score.

Finally, return the sum of all scores in the record.

For example:

ops = "52CD+"

- '5' - add to the record. Record now is [5]
- '2' - add to the record. Record now is [5,2]
- 'C' - invalid the previous score (2). Record now is [5]
- 'D' - Record new score that is double of previous score (5*2). Record now is [5,10]
- '+' - Record a new score that is the sum of the previous two scores. Record now is [5,10,15]

Return the sum: 5+10+15 = 30

For example:

Test	Result
cout << baseballScore("52CD+");	30
cout << baseballScore("524CD9++");	55

Answer: (penalty regime: 0 %)

Reset answer

```

1  int baseballScore(string ops){
2      /*TODO*/
3      vector<int> v;
4      int size=ops.size();
5      for(int i=0;i<size;i++){
6          //cout<<ops[i]<<endl;
7          if('0'<=ops[i]&&ops[i]<='9'){
8              v.push_back(ops[i]-48);
9              //cout<<ops[i]<<endl;
10         }
11         else if(ops[i]=='C'){
12             v.pop_back();
13         }
14         else if(ops[i]=='D'){
15             int temp=v.back();
16             //cout<<v.back()<<endl;
17             v.push_back(temp*2);
18         }
19         else if(ops[i]=='+'){
20             int temp=v.back();
21             vector<int>::iterator temp_back=v.end();
22             temp_back--;
23             temp_back--;
24             int temp2=*temp_back;
```

```
25         v.push_back(temp+temp2);
26     }
27     //cout<<v[0]<<endl;
28 }
29 int sum=0;
30 for(unsigned int i=0;i<v.size();i++){
31     sum+= v[i];
32 }
33 return sum;
34 }
```

	Test	Expected	Got	
✓	cout << baseballScore("52CD+");	30	30	✓
✓	cout << baseballScore("524CD9++");	55	55	✓
✓	cout << baseballScore("5C4C2C11+D3");	11	11	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3

Correct

Mark 1.00 out of 1.00

Given an array of positive integers `nums`. An array `mt` represents a mountain having no valleys and taking `nums` as its upperbound. In the other word, for all index `i` in range, `mt[i] ≤ nums[i]` and no pair of indices (`j`, `k`) that `j < i < k` and `mt[j] > mt[i] && mt[i] < mt[k]` exists.

Your task is to implement a function with following prototype:

```
int mountainWithoutValley(vector<int>& nums);
```

The function returns the maximum sum of numbers of `mt`.

Note:

- The `iostream`, `vector`, `climits` and `stack` libraries have been included and `namespace std` is being used. No other libraries are allowed.
- You can write helper functions.

For example:

Test	Result
vector<int> nums {10, 6, 8, 8}; cout << mountainWithoutValley(nums);	28

Answer: (penalty regime: 0 %)

Reset answer

```

1  int mountainWithoutValley(vector<int>& nums) {
2      // STUDENT ANSWER
3      vector<pair<int,int>> up;
4      vector<pair<int,int>> down;
5      int max=INT_MIN;
6      up.push_back({nums[0],0});
7      //down.push_back({nums[0],0});
8      int size=nums.size();
9      int sum1=nums[0];
10     for(int i=1;i<size;i++){
11         pair<int,int> temp_up_back=up.back();
12         if(nums[i]>=temp_up_back.first){
13             up.push_back({nums[i],i});
14             sum1+=nums[i];
15         }
16         else {
17             int sum2;
18             if(nums[i]*(size-i-1)+sum1<=max){
19                 goto skip;
20             }
21             down.push_back({nums[i],i});
22             sum2=nums[i];
23             for(int j=i+1;j<size;j++){
24                 pair<int,int> temp_down_back=down.back();
25                 if(nums[j]<=temp_down_back.first){
26                     if(sum1+sum2+nums[j]*(size-j-1)<max) {down.clear();goto skip;}
27                     down.push_back({nums[j],j});
28                     sum2+=nums[j];
29                 }
30                 else{
31                     //temp_down_back.second++;
32                     down[down.size()-1].second++;
33                     sum2+=down[down.size()-1].first;
34                 }
35             }
36             if(sum1+sum2>max){

```

```
37         max=sum1+sum2;
38     }
39     down.clear();
40     skip:
41     while(nums[i]<temp_up_back.first){
42         if(up.size()==1){sum1-=up[0].first*(up[0].second+1);}
43         else {
44             int up_size=up.size();
45             sum1 -= up[up_size-1].first*(up[up_size-1].second-up[up_size-2].second);
46
47         }
48         //sum1-=temp_up_back.first;
49         up.pop_back();
50         if(up.size()==0){
51             break;
52         }
```

	Test	Expected	Got	
✓	vector<int> nums {10, 6, 8, 8}; cout << mountainWithoutValley(nums);	28	28	✓
✓	vector<int> nums {163}; cout << mountainWithoutValley(nums);	163	163	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 1.00 out of 1.00

Given an array `nums[]` of size `N` having distinct elements, the task is to find the next greater element for each element of the array

Next greater element of an element in the array is the nearest element on the right which is greater than the current element.

If there does not exist a next greater of a element, the next greater element for it is `-1`

Note: `iostream`, `stack` and `vector` are already included

Constraints:

$1 \leq \text{nums.length} \leq 10^5$

$0 \leq \text{nums}[i] \leq 10^9$

Example 1:

Input:

`nums = {15, 2, 4, 10}`

Output:

`{-1, 4, 10, -1}`

Example 2:

Input:

`nums = {1, 4, 6, 9, 6}`

Output:

`{4, 6, 9, -1, -1}`

For example:

Test	Input	Result
<pre>int N; cin >> N; vector<int> nums(N); for(int i = 0; i < N; i++) cin >> nums[i]; vector<int> greaterNums = nextGreater(nums); for(int i : greaterNums) cout << i << ' '; cout << '\n';</pre>	<pre>4 15 2 4 10</pre>	<pre>-1 4 10 -1</pre>
<pre>int N; cin >> N; vector<int> nums(N); for(int i = 0; i < N; i++) cin >> nums[i]; vector<int> greaterNums = nextGreater(nums); for(int i : greaterNums) cout << i << ' '; cout << '\n';</pre>	<pre>5 1 4 6 9 6</pre>	<pre>4 6 9 -1 -1</pre>

Answer: (penalty regime: 0 %)

Reset answer

```
1 // iostream, stack and vector are included
2 vector<int> nextGreater(vector<int>& arr){
3     int n=arr.size();
4     vector<int>ret;
5     ret.resize(arr.size());
6     stack<pair<int,int>> s;
```

```

7   s.push({arr[0],0});
8   for(int i=1;i<n;i++){
9       pair<int,int> temp=s.top();
10      if(temp.first<arr[i]){
11          while(temp.first<arr[i]){
12              ret[temp.second]=arr[i];
13              s.pop();
14              if(s.size()==0){
15                  goto stop;
16              }
17              temp=s.top();
18          }
19          stop:
20          s.push({arr[i],i});
21      }
22      else {
23          s.push({arr[i],i});
24      }
25  }
26  int stackn=s.size();
27  for(int i=0;i<stackn;i++){
28      pair<int,int> temp=s.top();
29      ret[temp.second]=-1;
30      s.pop();
31  }
32  return ret;
33 }

```

	Test	Input	Expected	Got	
✓	<pre> int N; cin >> N; vector<int> nums(N); for(int i = 0; i < N; i++) cin >> nums[i]; vector<int> greaterNums = nextGreater(nums); for(int i : greaterNums) cout << i << ' '; cout << '\n'; </pre>	<pre> 4 15 2 4 10 </pre>	<pre> -1 4 10 -1 </pre>	<pre> -1 4 10 -1 </pre>	✓
✓	<pre> int N; cin >> N; vector<int> nums(N); for(int i = 0; i < N; i++) cin >> nums[i]; vector<int> greaterNums = nextGreater(nums); for(int i : greaterNums) cout << i << ' '; cout << '\n'; </pre>	<pre> 5 1 4 6 9 6 </pre>	<pre> 4 6 9 -1 -1 </pre>	<pre> 4 6 9 -1 -1 </pre>	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5

Correct

Mark 1.00 out of 1.00

Given a string consisting of 'a'-'z' and brackets '(' and ')'.
Your task is to implement a function with following prototype:

```
string parenthesesReversal(string s);
```

The function returns the string after recursively reversing every string enclosed between brackets.

Note:

- The `iostream`, `vector` and `stack` libraries have been included and `namespace std` is being used. No other libraries are allowed.
- You can write helper functions.

For example:

Test	Result
string s = "vi(ik)kq"; cout << parenthesesReversal(s);	vikikq

Answer: (penalty regime: 0 %)

Reset answer

```

1 stack<int> stac;
2 void rev(string& s, int start, int end);
3 string pararev(string &s, int i);
4 string parenthesesReversal(string s) {
5     // STUDENT ANSWER
6     return pararev(s, 0);
7 }
8 //void rev(string &s,int start,int end);
9 string pararev(string &s, int i) {
10     //cout << i << endl;
11     //cout << s[i] << endl;
12     //cout << stac.size()<<endl;
13     if (s[i] == '\0') {
14         return s;
15     }
16     if (s[i] == '(') {
17         //cout << "not ok" << endl;
18         stac.push(i);
19         s.erase(i, 1);
20         return pararev(s, i);
21     }
22     else if (s[i] == ')') {
23         rev(s, stac.top(), i - 1);
24         s.erase(i, 1);
25         stac.pop();
26         return pararev(s, i);
27     }
28     return pararev(s, i + 1);
29 }
30 void rev(string& s, int start, int end) {
31     for (int i = start, j = end; i < j; i++, j--) {
32         char temp = s[i];
33         s[i] = s[j];
34         s[j] = temp;
35     }
36 }
37 }
```

	Test	Expected	Got	
✓	<pre>string s = "vi(ik)kq"; cout << parenthesesReversal(s);</pre>	vikikq	vikikq	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



Question 6

Correct

Mark 1.00 out of 1.00

Given a string **S** of characters, a *duplicate removal* consists of choosing two adjacent and equal letters, and removing them.

We repeatedly make duplicate removals on **S** until we no longer can.

Return the final string after all such duplicate removals have been made.

Included libraries: vector, list, stack

For example:

Test	Result
cout << removeDuplicates("abbaca");	ca
cout << removeDuplicates("aab");	b

Answer: (penalty regime: 0 %)

Reset answer

```
1 string removeDuplicates(string S){
2     /*TODO*/
3     stack<char> stac;
4     stac.push(S[0]);
5     for(unsigned int i=1;i<S.size();i++){
6         if(stac.top()==S[i]){
7             while(stac.top()==S[i]){
8                 S.erase(i-1,2);
9                 i--;
10                //cout<<S[i]<<endl;
11                //cout<<stac.top()<<endl;
12                stac.pop();
13                if(stac.size()==0){
14                    break;
15                }
16            }
17            stac.push(S[i]);
18        }
19        else {
20            stac.push(S[i]);
21        }
22    }
23    return S;
24 }
```

	Test	Expected	Got	
✓	cout << removeDuplicates("abbaca");	ca	ca	✓
✓	cout << removeDuplicates("aab");	b	b	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



Question 7

Correct

Mark 1.00 out of 1.00

Given an array of integers.

Your task is to implement a function with following prototype:

```
int sumOfMinSubarray(vector<int>& nums);
```

The function returns the sum of the minimum value of every subarray of `nums`. The sum may be too large, so the result should mod `10000`.

Note:

- The `iostream`, `vector` and `stack` libraries have been included and `namespace std` is being used. No other libraries are allowed.
- You can write helper functions.

For example:

Test	Result
<pre>vector<int> nums {1, 2, 4, 3}; cout << sumOfMinSubarray(nums);</pre>	20

Answer: (penalty regime: 0 %)

Reset answer

```

1  int sumOfMinSubarray(vector<int>& a) {
2      // STUDENT ANSWER
3      int mod = 10000;
4      stack<int>st;
5      int n = a.size();
6      int dp[n];
7      for (int i =0; i<n; i++)
8      {
9          while (!st.empty()&&a[st.top()]>=a[i])
10         {
11             st.pop();
12         }
13         if (!st.empty())
14         {
15             int pre = st.top();
16             dp[i] = dp[pre]+(i-pre)*a[i];
17         }
18         else {
19             // or it doesn't exist, in this case the current element
20             // contributes with all subarrays ending at i
21             dp[i] = (i + 1) * a[i];
22         }
23         // push the current index
24         st.push(i);
25     }
26     int res = 0;
27     for (int i =0; i<n; i++)
28     {
29         res+=dp[i];
30         res%=mod;
31     }
32     return res;
33 }
```

	Test	Expected	Got	
✓	vector<int> nums {1, 2, 4, 3}; cout << sumOfMinSubarray(nums);	20	20	✓
✓	vector<int> nums {}; cout << sumOfMinSubarray(nums);	0	0	✓
✓	vector<int> nums {9830, 5335, 3573, 9754, 1778, 2960, 4386, 6461, 2524, 4734, 4263, 2571, 4628, 1981, 19, 4444, 5446, 3117, 102, 6044}; cout << sumOfMinSubarray(nums);	8583	8583	✓
✓	vector<int> nums {4921, 3462, 7784, 7764, 4246, 1659, 3629, 1594, 7885, 2625, 2038, 3252, 597, 7346, 4043, 1762, 2867, 8365, 7464, 461, 4238, 3441, 5814, 4218, 9008, 6826, 1084, 1956, 7199, 7882, 4550, 3241, 780, 2476, 7972, 7699, 203, 8101, 9334, 2539, 789, 2652, 1462, 6745, 619, 3147, 9558, 3230, 7339, 5126, 2589, 7669, 2584, 540, 4380, 8695, 9893, 3164, 2949, 159, 2082, 6138, 1582, 8605, 2682, 6193, 6802, 4079, 2492, 4091, 9689, 4733, 9881, 7519, 7658, 8803, 8925, 6603, 3950, 7012, 1426, 9346, 8566, 477, 6433, 8231, 2175, 9241, 1914, 9755, 4133, 1683, 2535, 3220, 7222, 9488, 1227, 3150, 697, 5822}; cout << sumOfMinSubarray(nums);	4631	4631	✓
✓	vector<int> nums {2544, 5222, 2719, 600, 5010, 8197, 3144, 8364, 4134, 3449, 2665, 3262, 9024, 922, 6669, 8798, 5315, 1660, 7497, 4695, 3977, 2544, 84, 4538, 1653, 6336, 8039, 1611, 7447, 1835, 3010, 2761, 7985, 3421, 8687, 5396, 1368, 9443, 1834, 3578, 8285, 9598, 1697, 499, 1068, 8006, 7865, 6222, 1108, 3376, 9185, 9499, 3720, 9571, 1483, 1482, 2118, 1799, 3912, 6292, 3878, 2831, 9555, 5017, 7717, 6329, 2241, 9977, 4579, 4276, 104, 9047, 233, 5601, 2045, 4539, 7231, 5309, 1746, 8327, 3581, 7806, 3856, 924, 3056, 1595, 401, 8957, 7264, 4401, 8034, 2148, 7328, 2211, 2006, 3620, 4069, 690, 5806, 8032, 9002, 6734, 5720, 8586, 2259, 9419, 6326, 2334, 5921, 3032, 3137, 7135, 9318, 6780, 9741, 9074, 4329, 1473, 3070, 5386, 12, 8037, 4139, 628, 572, 4303, 35, 8820, 1721, 655, 6813, 6079, 6039, 7417, 1684, 5227, 691, 884, 2163, 5363, 4282, 5166, 1021, 8352, 1095, 6770, 4544, 4117, 4155, 9244, 955, 8409, 8402, 8476, 9277, 4314, 5123, 7226, 1936, 80, 7704, 7785, 7079, 3471, 288, 9646, 2387, 5682, 2513, 7235, 492, 9012, 3729, 3959, 8542, 5200, 2967, 1657, 354, 125, 3422, 7045, 4143, 2114, 5594, 4430, 8272, 5856, 3997, 2892, 4007, 6079, 6940, 8279, 658, 7929, 7854, 3301, 1918, 1415, 2597, 8657, 1097, 424, 6112, 2685, 4077, 4585, 7951, 2729, 4265, 162, 6217, 2500, 5186, 2573, 9867, 2017, 7562, 5196, 8946, 1470, 8399, 1947, 9467, 2978, 456, 1700, 6344, 9285, 5756, 338, 2630, 8901, 6003, 3416, 3085, 1174, 4624, 9858, 3112, 377, 6688, 1196, 5365, 1488, 6332, 4002, 7390, 928, 5829, 3416, 5994, 1384, 2497, 2027, 2142, 129, 7152, 5399, 9336, 3991, 7381, 6811, 7725, 7432, 8849, 3562, 2273, 1978, 2418, 6355, 6167, 3473, 1340, 259, 730, 9217, 1222, 905, 3916, 4865, 601, 4426, 1596, 7617, 5076, 2647, 781, 8157, 4739, 5244, 4676, 371, 6129, 4410, 7042, 5585, 8284, 4807, 7268, 1442, 5544, 8628, 4423, 4459, 8023, 9983, 8144, 6384, 423, 4234, 315, 5565, 9948, 2204, 2017, 642, 4608, 3815, 2480, 4590, 520, 282, 6182, 2399, 3801, 5927, 3165, 2105, 6267, 6695, 3166, 5238, 92, 2141, 3071, 8716, 8776, 3064, 9406, 4709, 6968, 5853, 8115, 6011, 2488, 789, 3953, 7490, 9261, 7523, 4990, 4363, 2343, 6856, 9172, 6774, 4660, 3370, 3798, 375, 838, 6067, 2102, 6221, 9077, 2646, 5190, 735, 4487, 5031, 590, 6935, 8047, 2379, 2894, 2756, 7380, 893, 3908, 682, 1364, 6675, 1184, 2589, 4137, 2448, 7702, 5054, 8826, 5196, 3460, 8421, 581, 897, 3877, 6575, 1372, 1271, 677, 3482, 9357, 9778, 6502, 6831, 7782, 6874, 9186, 5295, 2633, 2220, 6050, 3762, 7404, 4463, 9054, 6251, 7605, 4601, 4566, 5984, 1652, 1843, 7329, 4440, 1857, 3690, 8099, 6742, 6615, 9432, 7316, 725, 8129, 1267, 687, 1956, 7221, 4387, 40, 7765, 1262, 5610, 7122, 6833, 1870, 2223, 6176, 6974, 7763, 3691, 9594, 7635, 5417, 4664, 5699, 1274, 2641, 855, 4808, 951, 3073, 2225, 3547, 4713, 6091, 3520, 5425, 5359, 7106, 3647, 7464, 1709, 5131, 1967, 1534, 8255, 2337, 8811, 782, 6557, 2939, 5694, 284, 9103, 3793, 2532, 8451, 867, 381, 559, 3852, 1693, 1471, 8427, 4626, 4030, 1844, 4467}; cout << sumOfMinSubarray(nums);	4396	4396	✓



	Test	Expected	Got	
✓	<p>vector<int> nums {7268, 9416, 2343, 2597, 282, 544, 1112, 563, 732, 4645, 4381, 1228, 688, 1441, 2006, 6651, 1170, 2206, 5685, 7998, 8888, 3642, 9678, 7641, 8278, 9710, 1163, 1564, 3696, 203, 4531, 3795, 8735, 6454, 7532, 8396, 2438, 5094, 4926, 7966, 4901, 3643, 4970, 1224, 4365, 5473, 7147, 8200, 5525, 854, 8620, 2829, 4364, 6360, 1834, 5935, 2640, 5074, 2268, 1613, 9022, 9401, 2144, 2327, 1083, 8130, 1431, 190, 2995, 3590, 7300, 9112, 2961, 2216, 769, 9769, 2638, 1606, 3859, 7582, 8124, 1679, 5358, 5243, 2885, 5957, 2540, 3176, 7456, 578, 762, 7767, 1221, 6895, 4916, 953, 7072, 1587, 5266, 8034, 1675, 1673, 578, 6234, 5335, 4517, 8137, 4277, 5915, 5736, 8180, 2101, 9713, 5106, 2820, 2671, 348, 1240, 2489, 9932, 287, 5574, 2318, 8432, 8794, 6438, 1925, 6160, 8198, 3298, 8884, 9503, 2244, 6854, 9345, 5466, 5648, 6794, 2058, 8614, 5260, 8398, 6696, 1894, 2538, 2582, 2085, 7297, 2233, 8425, 5691, 4066, 1893, 4901, 2303, 2930, 2600, 5349, 5566, 4500, 229, 2454, 7366, 319, 1895, 4054, 9800, 4706, 4565, 781, 5517, 1394, 6856, 4769, 2071, 2695, 7584, 9349, 1161, 9448, 6372, 6673, 5001, 1812, 7305, 7192, 3013, 868, 1382, 3893, 3236, 8125, 2133, 9272, 9951, 3721, 7403, 5600, 383, 216, 1484, 2009, 121, 2147, 6473, 4664, 4743, 3074, 7339, 3364, 6056, 6974, 1253, 8712, 5516, 9465, 7525, 4870, 3672, 2765, 699, 6974, 1185, 2019, 1613, 7759, 2807, 6629, 6202, 8461, 1151, 6630, 1773, 4260, 7174, 2384, 7765, 8076, 4002, 9965, 3027, 4685, 3099, 8634, 7715, 5749, 1179, 1690, 5055, 8616, 4897, 1127, 2304, 4135, 8972, 6766, 6182, 5716, 9030, 5179, 2538, 4245, 7167, 5874, 2306, 5111, 444, 2833, 2277, 6868, 4144, 2066, 4608, 3320, 2153, 5173, 1720, 2561, 6478, 445, 3904, 7365, 5304, 9369, 1997, 2329, 8398, 3014, 3784, 9199, 1413, 4723, 8548, 4735, 9575, 2710, 1457, 393, 9556, 3437, 5601, 313, 5543, 1976, 3545, 5837, 6171, 9317, 1578, 2403, 6445, 8827, 1280, 3846, 2131, 8427, 9002, 3147, 2877, 7011, 3867, 9751, 4139, 7062, 4256, 3587, 64, 7497, 6340, 9731, 4901, 7822, 2289, 5091, 9121, 2358, 7271, 2595, 6104, 4707, 3489, 4868, 2193, 7228, 1430, 4579, 2316, 5685, 9358, 5590, 9139, 7838, 7383, 8245, 6625, 8746, 2038, 1867, 6401, 3427, 33, 1004, 2728, 4363, 940, 2521, 9109, 5942, 258, 9442, 4296, 9735, 8775, 9724, 9841, 8873, 7340, 1571, 9722, 2207, 9763, 6065, 6152, 173, 4019, 1040, 9677, 4472, 6622, 8044, 4130, 2059, 2657, 349, 1748, 2348, 811, 5796, 3052, 6591, 7410, 3251, 2787, 6474, 8339, 1378, 4357, 2584, 5192, 623, 7910, 3809, 6495, 85, 1730, 5993, 2200, 1057, 1033, 3688, 623, 2465, 9340, 2251, 4763, 5997, 369, 8586, 7974, 1051, 9037, 2697, 671, 7454, 7778, 1746, 972, 1646, 1254, 2786, 979, 5790, 4598, 2755, 6491, 6106, 9771, 8411, 1854, 6703, 893, 7827, 1778, 2880, 4197, 2922, 3945, 3365, 5623, 8027, 4693, 9063, 2928, 3788, 720, 970, 7932, 520, 3355, 4218, 6587, 6736, 8357, 7689, 7204, 2221, 6271, 7223, 9179, 602, 1645, 7411, 3616, 9231, 7407, 7217, 4184, 9664, 6529, 9965, 2543, 5912, 48, 2479, 4973, 9456, 1438, 6512, 537, 821, 1752, 5723, 2492, 1753, 790, 2423, 7304, 5672, 242, 8404, 2168, 3037, 8973, 6404, 4953, 9462, 8254, 5277, 5322, 7062, 4812, 6028, 2016, 2857, 5471, 1356, 2125, 9357, 3745, 2279, 882, 3280, 315, 1900, 8882, 7988, 7709, 9889, 3401, 250, 7934, 4795, 7036, 2862, 5509, 5164, 8659, 4831, 9111, 4268, 1413, 3602, 5362, 5579, 2930, 2535, 8585, 5925, 8487, 1516, 7297, 8568, 2878, 1584, 5547, 1426, 8980, 3963, 52, 1007, 6463, 6780, 8457, 8945, 9855, 8980, 3359, 1728, 9875, 5329, 8193, 6037, 5104, 5069, 715, 6433, 5566, 1224, 8295, 5523, 7636, 9555, 7718, 5626, 133, 2031, 9074, 9320, 9322, 1029, 1953, 4952, 8850, 3607, 1580, 2513, 2775, 1430, 7201, 9626, 7907, 166, 3301, 7073, 6819, 2847, 7678, 4663, 4363, 5494, 4161, 9662, 4762, 1627, 2135, 7494, 5952, 6740, 406, 6308, 6989, 602, 4589, 1924, 5387, 7217, 6655, 97, 8494, 4782, 6612, 642, 4523, 7765, 2892, 4039, 3063, 1153, 3493, 4596, 1761, 5591, 6399, 5261, 67, 7021, 6414, 4772, 565, 3206, 8640, 2222, 5139, 6929, 7872, 1452, 2289, 1620, 922, 8169, 7483, 7507, 3743, 5887, 5649 ...snip... 670, 7762, 7544, 4355, 1273, 2469, 357, 4524, 8147, 7692, 2760, 6750, 4185, 8002, 1741, 1751, 2325, 1418, 8062, 4715, 5081, 3440, 6772, 971, 1585, 9568, 2015, 7193, 1512, 2535, 1160, 9707, 2334, 8487, 757, 2590, 7816, 5758, 1773, 796, 9576, 9656, 2941, 2241, 863, 7092, 9050, 9611, 976, 3845, 9623, 5216, 5655, 9370, 1593, 6437, 4618, 5952, 7401, 2748, 2031, 3095, 5082, 4653, 6798, 2882, 8189, 3992, 6131, 1217, 4073, 4136, 9974, 6078, 349, 6296, 7948, 1428, 2853, 8936, 3675, 8079, 4953, 907, 3044, 2942, 2415, 7346, 3398, 8100, 2412, 6685, 3377, 5150, 3081, 3982, 5773, 1415, 954, 4657, 9980, 3241, 3803, 1050, 4077, 6836, 4203, 5901, 9746, 2308, 1070, 7315, 6837, 9373, 7678, 3308, 1846, 6858, 2370, 6496, 984, 5064, 5985, 2122, 9933, 6508, 8824, 1655, 5887, 9052, 8587, 1595, 145, 3676, 1305, 9014, 4693, 9687, 4115, 7557, 8111, 5777, 9566, 5003, 6553, 9259, 5398, 6722, 9608, 7271, 2709, 3804, 7855, 7980, 5365, 6383, 7459, 3884, 2050, 5361, 6480, 5684, 5796, 751, 1450, 1779, 9771, 5572, 149, 2303, 5260, 7769, 5113, 2741, 7212, 4200, 1648, 6687, 3914, 9173, 6610, 5995, 4481, 7087, 4292, 8041, 7933, 2739, 5404, 9893, 1407, 6722, 1944, 166, 5276, 4996, 5167, 1926, 5339, 8813, 9911, 3860, 2766, 5503, 7577, 1990, 8592, 3184, 3779, 2114, 49, 1150, 5116, 4431, 3374, 7472, 7869, 2407, 7636, 5854, 5198, 9504, 1877, 2372, 1199, 4556, 748, 2435, 1784, 5309, 8782, 136, 2434, 2269, 1091, 9425, 3240, 7437, 163, 8013, 200, 7500, 4420, 7300, 6607, 1867, 9715, 3863, 1764, 9742, 1136, 89, 9187, 5573, 1391, 7790, 7069, 2412, 1420, 9795, 3773, 6544, 9983, 1486, 79, 9715, 176, 6823, 1820, 8114, 3021, 8055, 1001, 773, 362, 1967, 9009, 2710, 1240, 1245, 1983, 7446, 4425, 9499, 1697, 540,</p>	8084	8084	✓

Test	Expected	Got
<pre> 6979, 9001, 9674, 4335, 291, 3500, 1751, 7063, 1764, 8118, 8036, 1582, 992, 2983, 5023, 5405, 5669, 3220, 6333, 8596, 7772, 9192, 6699, 954, 5862, 5145, 7368, 7190, 8984, 9532, 3506, 7501, 8617, 2571, 9198, 8589, 6922, 6902, 9388, 1721, 4357, 5832, 1562, 4758, 1064, 1819, 8231, 1349, 4020, 9502, 1947, 4675, 7569, 8870, 6038, 4013, 597, 4403, 2358, 6616, 85, 5333, 141, 9135, 8155, 1150, 6961, 478, 824, 3050, 2966, 2191, 7099, 3740, 8745, 7635, 643, 9706, 3278, 2036, 8626, 669, 1965, 8680, 8930, 1666, 9832, 8888, 9868, 1172, 6993, 9330, 8487, 1741, 5273, 6012, 2078, 2394, 4348, 6810, 3606, 9151, 1819, 801, 9502, 7035, 3273, 109, 6535, 4413, 9515, 6976, 4678, 6567, 5034, 3484, 7591, 6866, 107, 2163, 5508, 2611, 5841, 5981, 4279, 524, 96, 3818, 7053, 2806, 2805, 360, 9595, 9341, 661, 9463, 5717, 3148, 4435, 8043, 3804, 6171, 1670, 802, 4776, 7590, 938, 1767, 2007, 6224, 8101, 6614, 6757, 7496, 4605, 416, 268, 2075, 2249, 593, 1613, 9434, 4917, 9873, 6912, 7453, 4490, 7976, 7305, 1127, 1581, 6038, 4076, 2140, 4759, 4962, 8929, 7441, 6757, 2176, 9963, 115, 1504, 9064, 7494, 9611, 9055, 3525, 1359, 7552, 284, 2267, 4473, 2638, 565, 1133, 4701, 4246, 9220, 1262, 465, 565, 6897, 2966, 1692, 3641, 7478, 4327, 1785, 7601, 5171, 6561, 4730, 9684, 9987, 6994, 6239, 1589, 3326, 6626, 6872, 3974, 3041, 1583, 3447, 7787, 5227, 8660, 5897, 7408, 9978, 3823, 6038, 8829, 8165, 6435, 7558, 1688, 7739, 3355, 9863, 5628, 4460, 9344, 605, 9162, 4134, 3005, 8921, 3837, 6083, 848, 3758, 818, 272, 7476, 2063, 1863, 4449, 173, 8924, 8787, 3876, 3588, 5039, 9367, 8771, 7252, 834, 1460, 7134, 4388, 9860, 2397, 5736, 2614, 2908, 781, 2048, 2047, 6108, 7308, 9892, 7751, 8921, 7262, 945, 5699, 186, 9629, 2142, 434, 6000, 9928, 1705, 5301, 357, 6172, 6626, 3267, 1704, 594, 2007, 9355, 8647, 9321, 2943, 5499, 3518, 5398, 5515, 1424, 5775, 9032, 1850, 3167, 5826, 1105, 3244, 7124, 624, 9154, 5050, 169, 362, 2070, 9011, 1594, 5056, 6547, 6314, 1062, 4142, 8124, 7918, 8072, 9433, 8362, 90, 3094, 7035, 9099, 5534, 6330, 539, 9356, 7421, 4432, 6193, 8147, 7338, 511, 824, 9056, 2748, 1611, 6277, 7846, 2249, 218, 1303, 1108, 1410, 5958, 383, 2777, 1339, 8242, 5602, 4791, 9551, 2001, 4620, 9367, 9553, 9843, 424, 3629, 2134, 4849, 2765, 45, 1537, 9739, 7339, 8494, 2110}; cout << sumOfMinSubarray(nums); </pre>		



	Test	Expected	Got	
✓	<p>vector<int> nums {2960, 9031, 2351, 9859, 4046, 1442, 5479, 2164, 9243, 3681, 2977, 3495, 4387, 713, 9960, 3767, 7554, 5689, 3375, 7004, 1860, 93, 6053, 9883, 9227, 9311, 6454, 7210, 8274, 9664, 5477, 1143, 6688, 4936, 4841, 2266, 5534, 5227, 6901, 6119, 9244, 8317, 1309, 4824, 7217, 198, 230, 3547, 6594, 3449, 5229, 3768, 2933, 1618, 1371, 4630, 1110, 622, 5655, 1102, 7466, 4704, 5788, 3455, 4002, 8561, 1500, 804, 562, 7047, 389, 883, 4723, 2362, 7284, 6680, 8725, 4131, 8040, 6993, 1929, 9365, 3192, 4939, 8734, 1712, 7639, 3165, 5546, 5283, 1158, 7475, 3788, 1235, 5755, 4414, 2080, 8266, 977, 1562, 1826, 5937, 4272, 2432, 4246, 9376, 7342, 7933, 4327, 9989, 1910, 7811, 639, 3156, 7992, 4830, 1751, 8959, 690, 7882, 3551, 134, 2333, 1507, 8832, 2672, 7786, 6060, 2368, 666, 543, 9844, 2678, 8335, 7421, 6806, 6814, 6713, 7620, 8369, 4559, 2663, 2601, 5304, 5771, 3871, 4518, 1434, 663, 7654, 2434, 1235, 412, 3038, 9316, 3124, 6542, 9839, 2066, 4962, 5051, 1706, 9934, 9099, 6053, 5321, 2020, 1182, 1235, 8283, 1181, 2522, 9541, 1678, 2488, 1607, 1879, 3278, 7353, 7493, 968, 1198, 5330, 6818, 7387, 116, 8097, 5936, 5981, 3737, 4858, 3286, 9048, 4513, 5815, 5829, 2776, 5384, 797, 4494, 3981, 7950, 5249, 8610, 7280, 1205, 1723, 8922, 2249, 6936, 2473, 9432, 6048, 398, 5143, 9982, 7234, 4047, 7134, 8302, 6870, 7238, 2971, 8801, 2017, 7080, 4545, 1916, 6023, 1652, 2561, 9583, 1331, 6139, 9513, 6832, 8137, 2818, 3770, 5048, 8906, 4197, 5551, 5660, 1639, 703, 955, 9983, 4218, 5768, 1810, 9494, 2977, 621, 9291, 9340, 348, 4582, 1108, 6758, 6858, 4660, 6663, 6586, 2767, 9834, 2284, 1068, 360, 5501, 2638, 617, 914, 2671, 7490, 1032, 5831, 4968, 9621, 605, 8570, 7238, 7206, 767, 1494, 3271, 8948, 1101, 4706, 7060, 8489, 6205, 8696, 9585, 231, 1237, 2039, 3202, 7106, 9950, 7317, 112, 8414, 4629, 5000, 987, 2282, 9619, 135, 3458, 6846, 2304, 2028, 8674, 9793, 2595, 9197, 3052, 8686, 7581, 1723, 4506, 5527, 9784, 75, 6242, 6474, 9338, 8868, 8043, 1683, 5790, 1959, 448, 8963, 3380, 4934, 6036, 1920, 4711, 5646, 8309, 573, 1760, 5062, 6851, 2971, 4665, 749, 6930, 5791, 645, 2303, 3471, 1830, 9006, 4649, 4624, 1911, 5055, 7245, 575, 648, 2884, 5128, 6647, 5460, 8766, 5986, 1263, 2555, 3275, 5813, 1832, 3130, 2325, 9650, 4034, 8140, 345, 3168, 3748, 5329, 2220, 7782, 1531, 95, 4601, 8427, 9362, 6374, 5390, 4364, 1875, 8328, 9208, 2782, 1789, 5161, 2747, 9397, 5607, 7267, 7031, 9204, 8212, 1853, 4647, 6502, 6279, 9789, 7756, 1356, 5288, 1212, 9833, 9303, 1751, 8249, 6512, 8073, 4193, 4704, 174, 4455, 2978, 3212, 1954, 6002, 3594, 330, 8419, 5143, 4562, 3297, 4147, 5043, 1808, 4623, 4974, 5066, 665, 7637, 5928, 9802, 5207, 2358, 6614, 1212, 3755, 1148, 1131, 7067, 8565, 3356, 6534, 4951, 7010, 715, 5834, 988, 4420, 4848, 6175, 4703, 1880, 347, 9239, 2137, 1621, 5399, 5155, 9064, 5297, 5565, 1414, 3650, 7336, 3769, 218, 2123, 959, 576, 1298, 2312, 3797, 1026, 5504, 7556, 4283, 3181, 936, 3524, 4038, 2066, 9237, 8295, 6803, 2284, 6139, 1648, 1290, 8685, 1564, 2646, 9302, 9915, 8910, 7627, 4612, 3756, 4973, 6620, 6729, 2478, 1837, 4761, 5849, 8027, 2643, 9942, 9974, 5418, 2101, 49, 4086, 2488, 1966, 405, 664, 7129, 7895, 5390, 6328, 2298, 7250, 7006, 7297, 8425, 3401, 4351, 2914, 1176, 6967, 8025, 3764, 2493, 4020, 8790, 8091, 9677, 4647, 1285, 1979, 3627, 3383, 9356, 1917, 6091, 483, 8722, 7521, 908, 6771, 7532, 726, 7365, 7022, 4838, 3225, 5759, 8013, 1556, 9995, 5428, 656, 3929, 5782, 3190, 1103, 1127, 8528, 147, 2740, 2264, 9217, 3703, 8056, 5358, 5682, 1698, 4810, 2438, 2454, 2732, 1479, 4594, 9278, 22, 4054, 9270, 9287, 9043, 7860, 6563, 1014, 1992, 6096, 659, 4760, 4063, 7255, 852, 5347, 5842, 2050, 9529, 4401, 8470, 302, 4317, 4782, 5909, 2269, 2249, 7103, 9713, 2715, 3751, 2485, 790, 7124, 3240, 3171, 9567, 36, 7348, 5772, 4124, 639, 7744, 929, 1539, 4162, 4933, 5834, 7556, 2233, 3998, 1266, 7681, 4595, 3153, 7720, 6941, 5779, 354, 8595, 356, 2088, 8997, 8261, 2428, 5071, 2883, 1892, 11, 2551, 1470, 1477, 1161, 5, 4520, 3849, 8967, 4425, 54 ...snip... 4710, 1977, 9530, 1656, 5377, 4815, 5177, 7643, 3824, 4819, 9860, 8195, 3379, 6425, 5285, 1868, 3716, 1589, 7339, 1034, 2935, 1029, 1789, 2714, 4644, 484, 8748, 1417, 1271, 4237, 1512, 1987, 8040, 61, 7734, 462, 6695, 7212, 5937, 5400, 7706, 3308, 1503, 8479, 651, 35, 9429, 320, 4891, 1366, 4141, 6225, 9644, 4695, 7046, 6531, 1069, 3025, 5965, 6258, 4422, 1552, 1579, 4943, 2279, 8775, 3343, 6224, 2861, 3916, 8926, 9318, 101, 6358, 696, 9023, 5635, 8858, 2253, 4861, 341, 242, 4047, 2600, 7923, 4896, 1097, 5375, 1578, 1970, 145, 9988, 7114, 7158, 5418, 1113, 5437, 6751, 6509, 5605, 9929, 488, 2888, 1355, 6087, 2009, 3208, 717, 2788, 8344, 2212, 1085, 7016, 3857, 1097, 2389, 2662, 1321, 7533, 2200, 6579, 447, 7326, 2679, 91, 8709, 4411, 696, 640, 1102, 694, 1941, 9924, 3512, 3496, 1617, 9705, 5707, 5349, 2816, 1910, 6658, 1922, 6567, 1453, 2490, 8180, 1729, 8555, 1913, 6874, 8100, 9904, 8994, 6862, 1700, 5600, 7653, 5176, 7574, 3369, 2507, 9908, 328, 6373, 595, 8885, 3263, 9042, 3079, 9514, 1872, 465, 2995, 3695, 7027, 8076, 8294, 3191, 1285, 7963, 6713, 3801, 1693, 2201, 2784, 8649, 8321, 4626, 9231, 8488, 378, 4223, 128, 4238, 2534, 1542, 370, 3047, 4828, 76, 1619, 3100, 7269, 188, 5873, 9540, 5705, 6429, 6365, 2874, 4551, 3054, 8481, 8225, 2506, 2005, 1636, 3125, 5576, 2425, 4171, 9284, 1420, 1692, 9458, 1549, 4855, 1965, 5260, 743, 2370, 3901, 6231, 5125, 6759, 1553, 3764, 2856, 9521, 8407, 902, 1750, 3393, 6428, 8232, 1450, 7586, 5741, 2771, 9580, 399, 7453, 6992, 4963, 2228, 3847, 8939, 3016, 6419, 4612, 410, 8124, 2856, 6734, 6461, 9612, 8158, 9302, 4469, 6740, 1843, 6563, 6281, 418, 2505, 493, 6311, 8941, 9359, 6649, 3592, 2310, 4256, 9669, 471,</p>	7671	7671	✓

Test	Expected	Got
<pre> 7085, 6314, 9415, 6115, 5895, 4722, 3489, 3059, 2734, 5238, 9911, 9258, 1130, 4882, 1380, 4044, 6751, 791, 7498, 1519, 1718, 9417, 8088, 5781, 8625, 536, 1393, 601, 1689, 6978, 539, 6711, 6906, 870, 3941, 1474, 5218, 92, 1340, 3168, 7350, 451, 7120, 6216, 8629, 3179, 7235, 4032, 4305, 933, 4002, 3040, 5643, 1720, 6326, 9035, 5358, 277, 8989, 7710, 1407, 3997, 3029, 7769, 9843, 2305, 6393, 7375, 9330, 153, 9647, 4489, 8906, 3113, 9936, 6335, 7767, 9362, 586, 8073, 1586, 517, 5683, 9139, 4424, 2606, 3823, 3952, 924, 5886, 5268, 9008, 2927, 8040, 540, 2261, 8239, 6241, 5322, 818, 7118, 898, 8625, 1797, 4037, 4287, 9521, 3255, 5816, 877, 8355, 9761, 6635, 2544, 39, 8289, 3605, 3634, 5163, 6669, 8500, 7723, 2255, 455, 3794, 205, 9048, 9826, 4135, 287, 4402, 6212, 782, 3917, 7225, 893, 764, 2558, 7689, 8023, 590, 6934, 8964, 2013, 9955, 6460, 6916, 3601, 2043, 894, 3666, 2649, 9261, 7060, 8791, 5418, 9327, 1651, 4960, 1128, 6497, 5732, 2423, 8141, 1371, 868, 8672, 3064, 9577, 4652, 8609, 6327, 451, 3411, 2342, 472, 6210, 5395, 6819, 2828, 4672, 1609, 6924, 77, 240, 1617, 1601, 1906, 3694, 1906, 6190, 5089, 8455, 3078, 530, 292, 9404, 9005, 2158, 1695, 5123, 2050, 8801, 8496, 7071, 718, 9921, 2872, 5297, 7227, 4300, 1129, 8671, 3067, 1264, 6465, 6467, 1099, 2536, 5334, 8839, 5026, 4920, 1650, 5601, 2047, 523, 3762, 9785, 2882, 8379, 2889, 4514, 9912, 2672, 5034, 2664, 4861, 9100, 4133, 2967, 3518, 103, 1073, 5597, 2621, 8983, 9276, 1601, 1656, 5795, 8527, 7251, 8971, 3767, 86, 7685, 1573, 6554, 2590, 9659, 953, 3998, 6543, 3487, 919, 6984, 8316, 8296, 4682, 3249, 4040, 204, 578, 4556, 4332, 9252, 1525, 3881, 5303, 6223, 1425, 3465, 6181, 7497, 8998, 7923, 6851, 8621, 9810, 2358, 7153, 2249, 3436, 9952, 808, 241, 3125, 3615, 8127, 7271, 4620, 8310, 2329, 9385, 8401, 338, 8289, 9515, 28, 1262, 2361, 8919, 4991, 9987, 399, 8690, 3193, 7476, 7729, 4784, 1356, 2329, 770, 9627, 4450, 5914, 7629, 2672, 4876, 5102, 1658, 1739, 1946, 5124, 2527, 8117, 6194, 8019, 9433, 9578, 2282, 5165, 3755, 1360, 3384, 2153, 693, 484, 8013, 5089, 2297, 9479, 4493, 3209, 5866, 3338, 5411, 7522, 7206, 69, 5426, 8029, 7974, 1630, 6766, 1571, 4723, 2632, 9612, 517, 1659, 5946, 5272, 8368, 9348, 450, 9622, 7411, 3985, 1385, 7211, 6628, 2850, 4633, 7788, 6863, 577, 9308, 3335}; cout << sumOfMinSubarray(nums); </pre>		



	Test	Expected	Got	
✓	<p>vector<int> nums {7456, 3530, 1415, 8785, 9636, 6314, 5719, 2015, 9803, 4492, 1627, 3529, 8641, 6272, 130, 6044, 2381, 7261, 2790, 400, 2793, 6296, 2137, 137, 9459, 8246, 2942, 721, 275, 6606, 5708, 8070, 1598, 1618, 2622, 7103, 2282, 559, 559, 3461, 2404, 8435, 724, 7568, 5550, 1983, 5258, 4480, 2989, 3870, 3602, 4397, 3218, 9365, 2330, 7281, 6919, 5594, 2959, 6452, 12, 1369, 9551, 9528, 4562, 7509, 9299, 2822, 3449, 9229, 477, 5240, 3461, 4176, 6942, 2265, 9526, 3841, 9467, 6058, 1944, 2110, 3345, 9511, 5959, 1261, 2102, 8121, 9383, 3032, 8466, 8429, 8836, 6924, 5631, 8757, 8005, 2732, 8445, 2061, 199, 8251, 1417, 6016, 4103, 3130, 2526, 140, 3967, 8142, 6972, 1860, 9733, 405, 3864, 85, 6176, 3003, 2133, 6043, 429, 5256, 1874, 1012, 3553, 7197, 7587, 4101, 6784, 2607, 2292, 4620, 253, 8194, 1563, 5590, 5890, 7737, 3886, 7393, 7394, 4749, 2533, 261, 8350, 3741, 6705, 6911, 8613, 7477, 1949, 7262, 2367, 9682, 3588, 5009, 8860, 6292, 1299, 2272, 7256, 1781, 4465, 5051, 6450, 7033, 3484, 6592, 7422, 5926, 1589, 4977, 679, 3703, 4223, 8004, 1289, 2587, 3107, 5533, 8670, 7747, 552, 5289, 7319, 7516, 9874, 8748, 7341, 7418, 1496, 4510, 7288, 2086, 5514, 7122, 8448, 153, 1987, 1719, 498, 2185, 5899, 290, 2582, 2451, 657, 4667, 8015, 3978, 5703, 2407, 1330, 5982, 4093, 4774, 5067, 3761, 2227, 5934, 8396, 9494, 8985, 3192, 3478, 7451, 5690, 5513, 6557, 2041, 2424, 6499, 1192, 9257, 6777, 9949, 4215, 313, 6531, 9889, 2264, 432, 6820, 8741, 6698, 4526, 965, 7660, 2435, 1671, 6303, 3325, 7616, 545, 4047, 6292, 8396, 4524, 2166, 9656, 6211, 2898, 4532, 2061, 4310, 911, 8342, 4250, 1809, 6787, 1549, 1678, 8203, 5493, 1250, 2191, 5535, 5898, 2871, 1491, 2217, 8295, 4687, 3058, 1874, 6003, 1344, 6316, 8880, 6512, 6280, 5431, 7960, 2028, 7518, 7899, 6165, 1938, 1643, 5474, 9485, 8775, 1496, 874, 2578, 2034, 2409, 4093, 8182, 1723, 7595, 9411, 8707, 3400, 6890, 4790, 7002, 8707, 3318, 6902, 507, 4880, 1831, 702, 1940, 3374, 2206, 8039, 6269, 923, 1886, 5115, 8179, 9517, 8235, 5613, 2563, 2879, 6178, 3303, 7338, 8264, 697, 4628, 4939, 8850, 4888, 1855, 112, 6138, 185, 1756, 356, 2115, 6779, 4245, 8945, 1056, 1661, 2104, 8406, 1938, 2281, 3887, 9483, 734, 1407, 5167, 9607, 7411, 2596, 9590, 6354, 7498, 3818, 3182, 8215, 9231, 9282, 9575, 4508, 415, 3775, 7412, 8709, 4498, 6013, 9589, 9349, 9371, 562, 8085, 631, 5614, 1891, 6181, 2877, 482, 3963, 6104, 8358, 6039, 4167, 1936, 933, 7737, 9141, 2249, 6272, 5685, 367, 4181, 310, 3835, 930, 6008, 116, 6565, 1348, 6708, 4186, 6183, 1365, 4040, 8665, 8072, 2245, 1140, 4656, 6992, 9399, 3361, 3272, 194, 2020, 4078, 9132, 7468, 9395, 766, 3885, 4844, 3671, 2192, 2757, 7797, 1801, 2511, 3402, 1107, 1528, 4796, 1302, 5067, 9839, 856, 804, 1991, 1798, 8822, 9005, 3137, 4523, 1936, 2005, 6797, 5282, 665, 4127, 9819, 618, 1571, 1800, 9901, 579, 8274, 4978, 9414, 9853, 4721, 5055, 244, 4994, 9491, 1970, 1472, 2295, 2253, 8544, 8308, 3166, 6802, 7996, 1964, 9456, 5198, 8180, 4863, 1705, 8328, 9750, 5327, 2164, 2033, 7799, 2304, 7706, 1842, 7989, 9366, 1861, 502, 4139, 6050, 7113, 5651, 4277, 1022, 8357, 319, 9349, 8982, 1430, 800, 6053, 2858, 4861, 1044, 5230, 593, 5970, 146, 3881, 1936, 3929, 3425, 7331, 8988, 292, 5992, 6854, 6812, 2377, 1622, 150, 2690, 4645, 7347, 6969, 8444, 961, 1670, 491, 1462, 3061, 4056, 7184, 3682, 7482, 8504, 9787, 5051, 7988, 1185, 591, 3925, 9119, 8777, 2884, 8985, 4345, 3036, 61, 7249, 4801, 1609, 6855, 4287, 2351, 7866, 1140, 1410, 820, 6826, 9673, 8059, 6995, 6258, 1589, 5869, 9072, 6676, 9987, 9252, 6295, 188, 3537, 6823, 4256, 2259, 8450, 4049, 8962, 38, 1217, 9116, 7542, 2359, 6759, 7055, 846, 6946, 239, 9465, 261, 835, 8872, 5502, 6328, 8006, 747, 6563, 230, 254, 9934, 772, 1892, 650, 5959, 9726, 6404, 870, 1221, 2489, 5702, 4974, 3515, 9958, 7855, 8600, 1102, 7162, 3210, 3538, 5602, 6793, 3853, 9881, 8800, 2615, 1085, 188, 2331, 2011, 2100, 6768, 8003, 2677, 9938, 9749, 4521, 1905, 5937, 1673, 5428, 2572, 8590, 7804, 5267, 262, 8005, 660, 2628, 2873, 2220, 5810, 917, ...snip... , 4943, 6829, 4804, 2446, 593, 2161, 5353, 9464, 1396, 3831, 8538, 189, 3166, 5550, 6153, 1302, 3287, 8203, 6197, 9377, 1152, 6479, 9009, 2124, 5556, 8630, 8451, 8914, 2334, 4272, 250, 6021, 1003, 5158, 9364, 766, 3249, 149, 6318, 6145, 2288, 5967, 7613, 9700, 1208, 5811, 5229, 4177, 1020, 7785, 9256, 969, 8934, 4434, 5738, 1756, 1338, 9775, 530, 9319, 5215, 9153, 827, 6621, 8891, 4289, 9883, 2713, 62, 6706, 3996, 9906, 5827, 9571, 4505, 9536, 3554, 9641, 2635, 4942, 7513, 7005, 1174, 9889, 4093, 2626, 1885, 4440, 8491, 9241, 4813, 2677, 7763, 8376, 5129, 3019, 1224, 104, 7537, 1255, 321, 6622, 7736, 2580, 9172, 1981, 1558, 4250, 7507, 5466, 4060, 4727, 3735, 80, 7494, 465, 3533, 6837, 2133, 7098, 2750, 3549, 2702, 4052, 2979, 4973, 5021, 583, 5625, 5154, 9470, 386, 3009, 8371, 7549, 638, 1099, 8900, 924, 2185, 8996, 8157, 309, 2066, 9501, 6827, 9729, 816, 5550, 997, 745, 4055, 7627, 9281, 3259, 2301, 8022, 8424, 7487, 3786, 5798, 6519, 4329, 5119, 6806, 9640, 3026, 729, 928, 6096, 2419, 2710, 8108, 1802, 1777, 5671, 2403, 7124, 895, 7318, 3673, 6552, 5408, 5978, 2715, 4159, 8125, 1768, 1445, 7425, 1862, 7276, 5991, 9647, 2268, 8729, 4553, 7332, 1646, 8605, 220, 2169, 1202, 680, 163, 2473, 6743, 8895, 6114, 6453, 2744, 5676, 2822, 8632, 3395, 2008, 917, 3197, 2139, 3356, 4263, 9888, 2604, 3108, 3765, 7282, 3869, 5510, 8785, 337, 3242, 2961, 7686, 4087, 4562, 280, 1873, 7154, 1374, 6961, 9313, 3124, 8532, 4314, 7611, 1580, 7099, 2092, 145, 7013, 1647, 1762, 4420, 173, 4823, 5211, 7223, 536, 2293, 6318, 662, 8454, 2766, 1598, 1511, 8883, 2363, 8414, 2415, 7581, 1423, 5270, 6760, 1051, 250, 2679, 9586, 8956, 4029, 188, 7160, 1822, 1727, 1633, 6788, 4624, 448,</p>	9382	9382	✓

	Test	Expected	Got
	<pre> 7688, 7782, 6122, 1132, 1962, 2112, 6235, 7522, 8027, 8726, 3651, 437, 5777, 9588, 1938, 530, 3862, 2064, 1441, 8627, 9548, 9986, 3536, 6820, 8288, 6497, 1135, 8965, 6946, 3168, 3265, 9295, 6516, 168, 8046, 8250, 8114, 3688, 7448, 8237, 9329, 9675, 6624, 3351, 6858, 6425, 394, 4687, 184, 8304, 6638, 5895, 3648, 8676, 7312, 3223, 6865, 8772, 9602, 5911, 9933, 2287, 872, 6299, 7847, 6887, 8685, 7478, 397, 2869, 6773, 9689, 5527, 2478, 3330, 7179, 7168, 2885, 4027, 5005, 789, 176, 6281, 254, 3036, 3519, 3188, 6109, 8263, 5146, 9166, 6621, 6617, 2570, 3134, 1518, 2472, 9291, 480, 6921, 8227, 6832, 3642, 76, 4054, 5800, 7666, 1191, 4940, 320, 5902, 486, 708, 153, 8380, 6944, 1255, 3771, 1588, 7935, 6697, 89, 7371, 6657, 1278, 1063, 7721, 7358, 288, 6506, 4237, 651, 2046, 2448, 3935, 1647, 4214, 9920, 8054, 4468, 501, 9778, 1068, 9712, 1310, 6785, 1006, 1636, 5642, 8384, 9384, 4581, 2096, 5852, 7361, 313, 9222, 2816, 4291, 4116, 9670, 8206, 5294, 4244, 8851, 5213, 2748, 5610, 5486, 126, 9592, 9290, 9668, 2004, 1019, 1924, 8603, 6402, 5064, 7044, 4557, 7703, 9897, 8743, 723, 760, 2344, 3391, 6774, 4269, 6178, 1672, 5330, 2911, 121, 4780, 8365, 5855, 4394, 4452, 9222, 2251, 6367, 9441, 9281, 4669, 933, 3178, 1538, 8644, 9824, 1846, 7383, 4535, 125, 3772, 6251, 9223, 1128, 4889, 1289, 8788, 3168, 3330, 9390, 4147, 792, 2539, 6866, 2031, 9631, 4087, 1416, 8206, 9138, 4672, 4435, 6837, 736, 2953, 939, 9230, 1855, 3459, 1761, 1384, 125, 2605, 5296, 5120, 5353, 8080, 1140, 2207, 2563, 9209, 4144, 4292, 1138, 2072, 6122, 20, 8436, 2777, 1001, 447, 3195, 4544, 6786, 2804, 4038, 315, 7807, 1656, 6093, 3571, 6354, 9379, 9794, 3299, 3577, 9483, 9492, 507, 5730, 8982, 2581, 4551, 6528, 729, 8894, 2204, 1694, 6870, 7234, 8371, 8788, 7510, 2193, 6282, 3480, 1027, 6430, 9446, 9478, 2730, 7538, 2605, 9155, 4879, 2082, 4564, 6990, 5585, 847, 7380, 5481, 8799, 7885, 2482, 2583, 6923, 4630, 2191, 5702, 5486, 7199, 2141, 7454, 2940, 9852, 1379, 986, 1354, 2720, 9651, 8999, 2573, 6791, 4186, 5339, 8404, 3623, 4696, 537, 2777, 5091, 6481, 1841, 5508, 659, 1251, 4061, 9649, 6209, 2793, 217, 1235, 7653, 4943, 2626, 1896, 2723, 8374, 5731, 1631, 8627, 2586, 1291, 1848, 4514, 8979, 2179, 7403, 7495, 4345, 2159, 3723, 2425, 8795, 52, 7116, 1231, 1458, 1817, 3459, 9863}}; cout << sumOfMinSubarray(nums); </pre>		



	Test	Expected	Got	
✓	<p>vector<int> nums {4577, 9408, 3645, 8203, 9984, 1798, 9011, 8496, 698, 3465, 8332, 161, 563, 5914, 1763, 2509, 6382, 2435, 3483, 3228, 7581, 4443, 6096, 4183, 4042, 8071, 1328, 7449, 8822, 9651, 4522, 1888, 3009, 7394, 4709, 6600, 1129, 2642, 3300, 6619, 4358, 5428, 7624, 1473, 5477, 2603, 2375, 4334, 8995, 2658, 3484, 7869, 2526, 4610, 1951, 1767, 3446, 3188, 5543, 841, 4185, 4933, 8476, 6948, 3359, 392, 5042, 8118, 4595, 1936, 8787, 7674, 1331, 9421, 2604, 8699, 7114, 5734, 1515, 7455, 9228, 425, 5639, 9378, 722, 4527, 48, 2176, 6967, 947, 1321, 2151, 3505, 6276, 2579, 9771, 9636, 4569, 492, 6172, 1715, 3861, 5576, 5078, 9561, 5964, 8830, 9960, 4840, 8788, 4791, 6631, 9920, 6031, 5106, 4931, 6065, 3464, 9068, 2473, 8836, 5148, 9918, 8571, 7863, 5867, 1863, 7918, 9048, 5422, 1067, 1698, 644, 8316, 1548, 5079, 8496, 7672, 7721, 4286, 1448, 957, 9428, 4989, 950, 4637, 3976, 920, 2037, 8101, 4394, 5324, 6486, 6149, 3983, 9277, 691, 1600, 4671, 4547, 5842, 3935, 3704, 2080, 1261, 5994, 3430, 3989, 1856, 7113, 8690, 8230, 1010, 3304, 7657, 2955, 5399, 1354, 6798, 2953, 5, 9759, 3026, 6224, 8853, 7033, 1395, 9893, 541, 4939, 2659, 2883, 236, 419, 2266, 5614, 9189, 6074, 2959, 6489, 3792, 6185, 2547, 9256, 9325, 1534, 7790, 2088, 619, 7855, 6750, 5854, 1992, 4398, 9741, 9037, 6377, 206, 8658, 5260, 1084, 8597, 8027, 4519, 3948, 2371, 314, 884, 9013, 4066, 7949, 89, 1033, 2270, 9795, 4045, 707, 1526, 5915, 8476, 2697, 4067, 5007, 895, 4071, 8153, 2625, 9387, 1002, 3565, 7509, 6664, 3881, 8854, 6288, 1109, 7435, 2150, 2799, 154, 2133, 1194, 9530, 2367, 7971, 3776, 7866, 4027, 8426, 5497, 6969, 6575, 6459, 943, 8602, 8564, 8506, 3497, 313, 9570, 7871, 8797, 1075, 7379, 2845, 2165, 6265, 7143, 9889, 2069, 7137, 6601, 5069, 7787, 6606, 3729, 3270, 5672, 8353, 1548, 7035, 4142, 895, 9209, 5463, 459, 9797, 5797, 8841, 1381, 282, 6410, 6401, 8614, 4948, 7123, 4237, 5163, 1790, 2707, 9238, 6535, 440, 3687, 7765, 8813, 4159, 5721, 7098, 8413, 5311, 934, 4510, 4494, 3718, 962, 7542, 1150, 5018, 1515, 7579, 1862, 2254, 2331, 7512, 3986, 8881, 2066, 2212, 3583, 1080, 723, 7681, 7141, 5677, 5789, 1444, 2183, 7719, 4413, 5590, 6248, 4188, 5731, 1178, 6529, 7541, 322, 3728, 1505, 7617, 3852, 8505, 9636, 7564, 5357, 5324, 6903, 4017, 358, 3273, 570, 4590, 9710, 3388, 342, 6583, 6500, 755, 2146, 6125, 4247, 9156, 8913, 2897, 8697, 2489, 9326, 6708, 306, 9325, 2107, 7872, 4625, 1596, 9574, 2556, 3576, 997, 5917, 2268, 6601, 9043, 9443, 8856, 2715, 6029, 2548, 5518, 4162, 5467, 1429, 8762, 2494, 4170, 5471, 3509, 3506, 6241, 2142, 4411, 600, 8373, 3493, 5761, 4867, 3642, 8420, 2423, 3445, 1659, 2301, 8635, 4610, 9035, 5395, 958, 9460, 4013, 9492, 4539, 5265, 2161, 9097, 1093, 6490, 5745, 475, 7792, 1349, 891, 1640, 7155, 7117, 3647, 8294, 5420, 7592, 2091, 328, 4596, 8714, 405, 6655, 6117, 7417, 5113, 1026, 4057, 1424, 4484, 9632, 2629, 3815, 3616, 8853, 822, 1939, 6032, 2268, 8906, 9866, 1711, 9125, 9219, 2442, 9428, 1863, 679, 2266, 1209, 9623, 8123, 1505, 7372, 2298, 9106, 1734, 1493, 6343, 6320, 1223, 6089, 752, 8468, 4734, 6655, 114, 2806, 7377, 4849, 7451, 2476, 377, 1861, 5004, 963, 9221, 8732, 6112, 2300, 7568, 8682, 9740, 2597, 4000, 8664, 9110, 8922, 1793, 7511, 3227, 7532, 1537, 6214, 3355, 9361, 461, 1525, 8632, 1371, 1404, 4586, 880, 1053, 8954, 5156, 5398, 3292, 7400, 543, 8440, 5203, 8626, 1784, 4284, 2617, 3985, 9121, 7996, 6582, 9970, 3158, 965, 2484, 7878, 977, 7274, 611, 1579, 2523, 334, 9695, 8220, 8068, 9506, 1350, 5349, 9461, 1067, 7653, 825, 884, 40, 4002, 2551, 137, 1506, 7167, 4842, 7854, 5348, 29, 3393, 8011, 9022, 6512, 4126, 771, 7829, 1682, 6194, 137, 8557, 8730, 4857, 2391, 7465, 2542, 8530, 3297, 9106, 8806, 642, 8981, 3127, 243, 7369, 6789, 4292, 7472, 4485, 4409, 4118, 1368, 5638, 6766, 5965, 6568, 8690, 9188, 1363, 3430, 3321, 4439, 8508, 8275, 1179, 619, 990, 7965, 9449, 302, 5092, 4824, 8058, 5560, 3709, 2397, 8623, 1606, 5706, 835, 7497, 904, 5513, 5166, 2265, 9239, 3307, 4613, 3880, 4091, 5244, 9176, 3960, 78 ...snip... 8, 3308, 1401, 3288, 8261, 4429, 9001, 1116, 1266, 5139, 3814, 8589, 2757, 1233, 8337, 6007, 9761, 8997, 7845, 4942, 9622, 5063, 4053, 9636, 5053, 1362, 2901, 7760, 5734, 4404, 7357, 6092, 2815, 4801, 3609, 2189, 9481, 5836, 2643, 4841, 7750, 8664, 21, 7123, 6565, 635, 3706, 2106, 81, 6660, 1525, 548, 1206, 543, 4884, 8485, 1014, 5751, 5238, 52, 139, 5324, 4394, 1280, 1262, 1909, 3642, 1902, 5848, 2263, 7512, 7543, 3622, 1046, 104, 7600, 2146, 2540, 7022, 9065, 6928, 5288, 1929, 3290, 5606, 8296, 1825, 1914, 1728, 1466, 685, 675, 403, 1663, 3880, 6851, 912, 8353, 1300, 7380, 1323, 8951, 6703, 446, 1260, 8438, 7190, 3290, 6181, 6171, 8807, 4138, 7798, 5494, 6475, 1880, 4878, 1417, 3780, 3101, 323, 6242, 8068, 7507, 3844, 5812, 8526, 7325, 8092, 6109, 6080, 1093, 7753, 1315, 465, 1446, 5614, 6320, 1335, 3915, 2166, 2624, 1197, 1879, 5083, 617, 9507, 7015, 662, 7643, 1270, 821, 6967, 4999, 1442, 2110, 5670, 340, 1589, 5590, 635, 5408, 7173, 9194, 2108, 4948, 5749, 4663, 2069, 5659, 7828, 8409, 4253, 405, 6870, 743, 3750, 551, 6760, 3069, 2485, 9349, 4286, 1145, 563, 4754, 1791, 794, 1747, 2455, 7939, 7366, 6805, 3043, 7931, 7052, 938, 1778, 6659, 889, 4745, 1216, 580, 2158, 4482, 8192, 9052, 3713, 1968, 8984, 1970, 6403, 9460, 9087, 4436, 8515, 1520, 8011, 1491, 8017, 9217, 1236, 9154, 1205, 4674, 6829, 8635, 2017, 1452, 3908, 9539, 980, 7570, 8775, 3199, 6993, 8940, 1001, 1511, 2143, 6642, 5093, 4729, 9826, 1355, 1283, 8337, 202, 3446, 1065, 5383, 8522, 4186, 5940, 3962, 2956, 8281, 71, 6492, 2355, 5978, 3618, 4046, 8625, 4903, 9607, 8990, 2266, 9908, 6793, 4093, 4680, 8946, 9078, 9862, 5360, 290, 4999, 1452, 6310, 6989, 8469, 9699, 5197, 9572, 3230, 561, 2369,</p>	5989	5989	✓

Test	Expected	Got
<pre> 7636, 9864, 8406, 1080, 3660, 5061, 5915, 7574, 5378, 4312, 5415, 8240, 6482, 8168, 7192, 1108, 3256, 2162, 3584, 1479, 3505, 2243, 9789, 9775, 5997, 8566, 5901, 1218, 648, 1643, 1537, 6493, 9785, 7097, 5442, 426, 4298, 7029, 8658, 9948, 3593, 6211, 6869, 1629, 2878, 5133, 4511, 4043, 990, 4898, 7343, 1406, 591, 4128, 7132, 6365, 8590, 7957, 6847, 2369, 6109, 9890, 1714, 5209, 2209, 8146, 1127, 3536, 5579, 5721, 4636, 7932, 8421, 5184, 205, 6664, 6735, 9568, 299, 629, 9530, 5021, 8443, 6508, 6262, 9174, 578, 216, 1623, 4265, 2320, 258, 1469, 8556, 1994, 6764, 4964, 5946, 7613, 8305, 2008, 4746, 8448, 2076, 1933, 2134, 905, 5721, 6975, 3531, 9346, 1914, 4998, 2364, 2189, 5675, 5884, 6076, 8838, 9992, 7921, 8907, 1801, 1869, 126, 5633, 9363, 6701, 3001, 8921, 2285, 1847, 8745, 5120, 6310, 230, 3243, 6222, 6912, 8696, 8708, 947, 2527, 6242, 5016, 2246, 2509, 4242, 8874, 2634, 7329, 8261, 7348, 1555, 1696, 6170, 3369, 3752, 3045, 8934, 1082, 2056, 6686, 4953, 534, 6962, 2215, 5841, 197, 4863, 5867, 6398, 3863, 5923, 8487, 325, 3176, 1012, 5793, 1225, 839, 9594, 5849, 2119, 200, 2904, 8112, 1345, 9302, 5721, 3878, 4670, 2701, 5823, 9288, 399, 5543, 568, 698, 7957, 1077, 4802, 6800, 1617, 2256, 7201, 4724, 5404, 112, 6030, 8131, 392, 1437, 9570, 1976, 7030, 9348, 7443, 5774, 330, 3917, 6222, 480, 9020, 7323, 4486, 2723, 177, 3219, 6411, 5143, 5689, 6737, 8271, 8441, 9130, 1591, 5397, 6509, 6621, 2423, 1227, 1466, 1959, 6514, 7050, 3830, 1291, 2562, 5147, 4644, 1437, 3862, 7340, 4778, 8858, 3148, 3331, 4856, 9959, 1972, 1299, 1174, 6519, 8465, 5757, 9214, 1857, 1923, 4072, 9658, 6417, 2676, 3252, 1279, 9822, 3200, 2235, 5610, 6028, 4648, 5321, 8841, 499, 5368, 5806, 2013, 2191, 5415, 8673, 2174, 4633, 4256, 7751, 3264, 5222, 1823, 7076, 8933, 5775, 125, 5890, 3300, 2833, 4944, 5312, 3368, 2629, 4813, 7196, 2060, 2435, 1919, 9988, 2181, 9794, 4252, 8522, 5200, 9859, 9798, 5078, 3630, 6607, 3184, 7392, 1964, 2174, 1105, 1998, 9880, 9466, 2030, 6458, 2642, 3971, 7571, 8872, 1535, 2708, 4257, 3110, 5700, 2838, 9423, 4965, 9873, 8052, 2522, 1496, 6203, 9998, 1775, 3675, 2334, 3666, 4219, 2021, 7299, 6119, 2113, 2384, 1263, 1804, 7515, 7721, 668, 9073, 4868, 6419, 6508, 7140, 6790, 6865, 667, 4095, 8176, 9860, 2356, 8751, 9002, 3681, 4662, 1880}; cout << sumOfMinSubarray(nums); </pre>		



	Test	Expected	Got	
✓	<p>vector<int> nums {4491, 1079, 1097, 1112, 7947, 2543, 3676, 3098, 1077, 2821, 3680, 8107, 4351, 102, 6443, 1906, 2653, 4234, 406, 6683, 1090, 7220, 3335, 6965, 3465, 9768, 8694, 6285, 4868, 3133, 762, 4842, 898, 4930, 2420, 7930, 70, 8342, 3279, 4749, 6806, 1682, 2965, 4609, 5425, 7594, 4038, 4605, 2708, 4085, 5045, 8547, 7119, 344, 8931, 3967, 3779, 3577, 3184, 4261, 7487, 851, 3534, 8282, 2684, 9430, 4665, 2321, 8645, 2369, 8836, 2777, 1307, 1138, 9110, 2293, 8744, 7132, 6373, 2140, 8323, 1531, 1937, 1922, 3795, 1508, 3220, 8759, 7415, 3969, 1513, 1038, 5508, 6939, 7019, 2737, 1969, 4264, 6959, 7235, 9616, 7752, 6388, 22, 5326, 8843, 3497, 4126, 7437, 293, 6912, 9542, 4287, 8692, 9523, 5218, 4087, 1697, 2516, 7828, 3098, 1877, 9065, 8658, 1004, 4447, 1335, 6040, 3448, 147, 7054, 3770, 1279, 7796, 5286, 9389, 3877, 6245, 2744, 3990, 6984, 8829, 2432, 2979, 4131, 7641, 402, 521, 1618, 3767, 6442, 7726, 1063, 3728, 8152, 8376, 5628, 5524, 7148, 1621, 2250, 4349, 5167, 7772, 9818, 4909, 5426, 1076, 5177, 8585, 7626, 4980, 7985, 8059, 252, 5915, 118, 2546, 4102, 7441, 2366, 5213, 7935, 605, 8581, 808, 8015, 5128, 1695, 5269, 8434, 5746, 5007, 6862, 2904, 4307, 1000, 1674, 17, 3653, 1714, 6591, 3080, 4944, 6160, 4552, 5469, 4646, 6015, 7526, 2615, 3729, 7425, 939, 474, 1449, 587, 8821, 4821, 4138, 1100, 5112, 2479, 7509, 2623, 4239, 7108, 2118, 1250, 5316, 8346, 4054, 4712, 2014, 9273, 5410, 5116, 8443, 7562, 1068, 1559, 6030, 6558, 9120, 3012, 490, 851, 3610, 4594, 5568, 1667, 6006, 858, 4965, 7117, 1608, 6434, 4056, 472, 9165, 6576, 467, 1941, 6513, 6899, 5935, 4997, 4354, 8046, 2813, 2715, 2677, 758, 9762, 1779, 5336, 7695, 818, 3634, 2200, 5778, 9873, 1344, 8852, 3106, 8309, 724, 5569, 351, 2144, 3694, 8160, 8106, 6398, 1179, 8093, 4202, 9385, 1508, 4504, 3262, 5381, 4264, 7033, 6356, 8681, 1052, 544, 5162, 3514, 5008, 9003, 6355, 511, 804, 4476, 7868, 1568, 4986, 9754, 5951, 8128, 1629, 5495, 7624, 4253, 9747, 1457, 2358, 2725, 4223, 2269, 2500, 8508, 7786, 4529, 9833, 2075, 5779, 6592, 9757, 347, 6164, 332, 4509, 3890, 6695, 6663, 7520, 8573, 7673, 4586, 1850, 7119, 526, 1242, 3729, 889, 1877, 4871, 7535, 1803, 8253, 503, 6008, 1961, 2168, 3923, 8636, 6417, 5359, 2848, 2639, 9420, 1606, 865, 5463, 836, 9826, 2875, 6789, 5029, 4328, 1625, 2632, 3558, 1864, 5761, 5679, 2266, 9577, 3894, 173, 3490, 102, 6815, 5761, 7381, 2378, 1272, 4487, 3043, 7817, 6824, 5625, 4550, 6383, 2579, 2106, 4267, 7932, 2561, 8293, 1279, 1040, 9727, 4385, 4605, 1283, 7556, 2312, 6215, 6472, 5222, 2216, 1219, 4275, 2494, 7785, 9352, 1161, 1001, 608, 8776, 3639, 3552, 115, 1425, 1622, 8587, 7134, 5359, 1213, 4386, 6998, 7844, 5512, 580, 1932, 1211, 5815, 4347, 357, 9049, 1194, 79, 8807, 3909, 3899, 810, 3222, 1561, 108, 7596, 3424, 5929, 243, 1745, 5397, 3140, 8995, 9296, 9211, 3545, 1262, 3460, 5761, 9884, 4920, 7008, 1603, 888, 493, 1374, 1661, 647, 2917, 3593, 9363, 4272, 1799, 1485, 3873, 641, 5819, 8379, 6398, 3374, 6437, 2097, 3111, 4664, 719, 3481, 5737, 9280, 2134, 1345, 9790, 7198, 5288, 3595, 5692, 3870, 1740, 2022, 3194, 3834, 2435, 8188, 2036, 6046, 4957, 685, 1121, 2327, 975, 1805, 8964, 3773, 6260, 3449, 2784, 1140, 7857, 9984, 5615, 9581, 9277, 4351, 9413, 6994, 8721, 1597, 768, 4178, 140, 1657, 1506, 6074, 11, 6353, 6482, 8453, 2823, 6564, 4240, 2532, 6031, 8089, 6520, 5281, 7912, 466, 9065, 5120, 945, 476, 6487, 4184, 604, 9704, 7309, 1121, 2330, 4030, 9366, 9631, 1510, 4864, 6574, 9470, 293, 8942, 8325, 45, 9004, 3721, 2084, 9039, 291, 8585, 8739, 9840, 985, 7650, 6186, 8262, 4904, 2228, 7495, 2152, 5418, 9379, 6815, 8835, 6539, 2279, 5528, 6298, 289, 5444, 2422, 3534, 2938, 2412, 7461, 2095, 4037, 3203, 4167, 2577, 836, 7476, 6797, 7651, 3072, 2563, 6267, 5164, 2429, 8114, 984, 8038, 8255, 9032, 851, 4031, 7853, 3326, 2028, 8510, 613, 3644, 6498, 9651, 913, 516, 6985, 5202, 921, 812, 2351, 210, 825, 1778, 7838, 9937, 387, 6443, 6548, 2298, 6760, 1069, 9105, 4649, 2276, 3534, 4689, 8014, 551, 4221, 3148, 7153, 9083, 8927, 5834, ...snip... 1671, 4741, 6, 2472, 6725, 4579, 5642, 3249, 407, 1047, 878, 8858, 95, 7199, 5685, 52, 6209, 4159, 2708, 8932, 5335, 9063, 6687, 1999, 7912, 115, 383, 1383, 9290, 2417, 9719, 4148, 4005, 6472, 9383, 4038, 3338, 7463, 4384, 7513, 4740, 1516, 2200, 7562, 1054, 4454, 9313, 6506, 886, 1389, 8454, 2801, 6352, 2032, 1445, 8324, 4029, 5791, 9491, 8202, 1486, 3051, 9180, 6892, 7363, 8394, 7770, 7051, 8473, 8960, 6457, 438, 4804, 3498, 6888, 3084, 3593, 2124, 7490, 8094, 5503, 6782, 2542, 4217, 8070, 6706, 1811, 9507, 5001, 777, 128, 1443, 1392, 5335, 5499, 7478, 3018, 8276, 2825, 3968, 2953, 2020, 2270, 9616, 9406, 2601, 2917, 1383, 166, 408, 3345, 6393, 1829, 8138, 6997, 6996, 6162, 3187, 9057, 8226, 7139, 8792, 8254, 346, 4395, 1308, 2445, 6188, 6624, 3948, 3706, 2732, 66, 7741, 8423, 8350, 5130, 3426, 5302, 8371, 9087, 8722, 9590, 387, 8334, 6089, 5347, 4284, 5995, 4637, 1271, 7337, 2835, 6388, 2405, 8566, 5572, 4996, 9975, 707, 6079, 1612, 8310, 3384, 7261, 4737, 27, 1816, 4640, 7243, 5028, 9289, 3899, 7394, 2361, 3699, 5109, 4098, 2623, 9174, 7097, 9591, 7196, 9873, 9489, 1253, 8780, 1856, 1397, 1334, 4868, 7830, 2591, 4762, 3269, 3148, 8913, 2538, 3924, 9832, 1145, 9854, 1815, 4865, 2307, 4272, 4610, 4593, 15, 4000, 9826, 6861, 8693, 1336, 5162, 5202, 6118, 2835, 3389, 6992, 805, 10, 9019, 59, 8982, 5264, 6719, 4111, 6326, 5423, 8232, 2388, 1164, 2657, 2423, 9618, 8942, 884, 7176, 8743, 9766, 4920, 6693, 2487, 6934, 9610, 4151, 8403, 3977, 4486, 2670, 3469, 6893, 9928, 9443, 3466, 4782, 7812, 8300, 5545, 7733, 1352, 4496, 1348, 8993, 9886, 1275, 1861, 8132, 5206, 6020, 6459, 6081, 1587, 9998, 524, 2132, 6132, 2732, 892, 5939, 6854, 104, 111, 9683, 3672, 3813, 5372,</p>	8789	8789	✓

Test	Expected	Got
<pre>6671, 4400, 851, 1588, 1061, 268, 1076, 5985, 2240, 4140, 2605, 2625, 5597, 7198, 2222, 5461, 1224, 4329, 9718, 6469, 7063, 6302, 4593, 9799, 9883, 6187, 5139, 3999, 8898, 3043, 3849, 3897, 6335, 923, 1486, 3557, 1166, 8565, 9755, 2979, 130, 8082, 6611, 5039, 9522, 8804, 1982, 878, 1844, 6306, 4564, 8538, 8664, 4736, 1309, 1353, 7001, 1566, 8726, 469, 5647, 4591, 7463, 2872, 3240, 5037, 8819, 2103, 7064, 9082, 9135, 881, 8591, 4442, 9462, 2183, 5018, 112, 1084, 6941, 1292, 743, 290, 1506, 798, 802, 6809, 9197, 7333, 2133, 555, 8648, 6276, 5781, 6614, 770, 1233, 1069, 4666, 1664, 4700, 6644, 8470, 1187, 7933, 4315, 4366, 928, 1357, 8731, 5967, 827, 4537, 6462, 3078, 3941, 4258, 2012, 142, 1609, 3160, 8170, 9048, 5921, 8520, 3294, 6688, 7934, 3635, 5109, 653, 6632, 5249, 2521, 9369, 1730, 6246, 1629, 498, 7474, 7508, 5947, 2657, 4352, 7089, 7889, 6153, 9901, 9531, 3398, 8242, 9813, 8377, 19, 1366, 6232, 2579, 8560, 2651, 4885, 5381, 1612, 6067, 3496, 5744, 9970, 5792, 6054, 2161, 7608, 5772, 742, 9366, 5033, 600, 1203, 7774, 6742, 8707, 7231, 1265, 904, 2093, 1370, 1888, 5322, 1709, 6244, 7278, 1156, 1824, 4015, 6415, 7866, 3, 1464, 5679, 5927, 9587, 4039, 391, 7380, 9602, 7908, 9755, 5710, 3293, 781, 3192, 8647, 1436, 4806, 3804, 5227, 1641, 4016, 5910, 5621, 1941, 1949, 6672, 947, 4951, 659, 5529, 6760, 4758, 9071, 7016, 6544, 7721, 1275, 3422, 3369, 7373, 5475, 5660, 4145, 4887, 3285, 6021, 986, 1003, 9752, 5653, 3509, 3645, 5662, 8738, 6990, 6501, 7596, 8977, 8708, 9592, 3721, 6524, 1244, 6900, 4805, 7571, 9958, 9516, 5229, 9825, 8064, 1509, 5436, 6528, 8489, 1888, 164, 2255, 9513, 8346, 2989, 5011, 9344, 9984, 557, 6444, 3868, 7494, 153, 9338, 4430, 1194, 3354, 4815, 2639, 8965, 6484, 390, 7983, 6642, 2467, 8709, 5087, 4866, 3575, 8735, 3344, 8686, 6433, 2893, 8347, 3135, 6462, 7006, 5538, 4891, 1428, 1180, 6294, 6832, 1451, 9826, 7323, 326, 3278, 8059, 4025, 2224, 178, 8219, 7189, 4196, 8038, 2500, 8127, 107, 6699, 7282, 2525, 2027, 2821, 552, 6136, 3779, 371, 927, 4117, 3254, 4572, 5643, 8838, 2395, 6043, 2209, 5810, 2676, 156, 9990, 7923, 4991, 5751, 9658, 2286, 4090, 3325, 1397, 937, 330, 7104, 7936, 6296, 3773, 9160, 5623, 2664, 189, 1475, 1293, 9087, 2312, 9229, 3073, 7203, 5263, 9504, 2415, 4763, 3516, 5293, 426, 1379}; cout << sumOfMinSubarray(nums);</pre>		

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



Question 8

Correct

Mark 1.00 out of 1.00

Given an array of integers.

Your task is to implement a function with following prototype:

```
bool tripleIndexSLM(vector<int>& nums);
```

The function returns whether a triplet of indices (i, j, k) such that $i < j < k$ and $nums[i] < nums[k] < nums[j]$ exists.

Note:

- The `iostream`, `vector` and `stack` libraries have been included and `namespace std` is being used. No other libraries are allowed.
- You can write helper functions.

For example:

Test	Result
vector<int> nums {1, 10, 0, 3, 3}; cout << tripleIndexSLM(nums);	1

Answer: (penalty regime: 0 %)

Reset answer

```

1 bool tripleIndexSLM(vector<int>& nums) {
2     // STUDENT ANSWER
3     if(nums.size()<=2){
4         return 0;
5     }
6     vector<int> v;
7     v.resize(2);
8     deque<int> de;
9     if(nums[0]<=nums[1]){
10        v[0]=nums[0];
11        v[1]=nums[1];
12    }
13    else{
14        v[0]=nums[1];
15        v[1]=nums[0];
16    }
17    for(unsigned int i=2;i<nums.size();i++){
18        if(v[0]<nums[i]&&nums[i]<v[1]){
19            return true;
20        }
21        if(nums[i]<=v[0]){
22            de.push_front(nums[i]);
23        }
24        else if(v[1]<=nums[i]){
25            de.push_back(nums[i]);
26        }
27        if(de.size()==2){
28            v[0]=de[0];
29            v[1]=de[1];
30            de.clear();
31        }
32    }
33    return false;
34 }
```

	Test	Expected	Got	
✓	vector<int> nums {1, 10, 0, 3, 3}; cout << tripleIndexSLM(nums);	1	1	✓
✓	vector<int> nums {}; cout << tripleIndexSLM(nums);	0	0	✓
✓	vector<int> nums {15815, 15258, 10488, 9802, 5861, 3992, 2481, 1331, 550, 539, -1392, -3572, -5180, -7837, -10398, -10764, -12797, -13580, -13724, -14400}; cout << tripleIndexSLM(nums);	0	0	✓
✓	vector<int> nums {15953, 15605, 15178, 14780, 14719, 14581, 14512, 14312, 13802, 13337, 13250, 13205, 13178, 12643, 12027, 11679, 11497, 10766, 10686, 10486, 10480, 10036, 9483, 9319, 9304, 8981, 8882, 8684, 8681, 8651, 8541, 8392, 8241, 8241, 8145, 7686, 7686, 7447, 7222, 7200, 6754, 6284, 5692, 4977, 4146, 3992, 3704, 3439, 3254, 2699, 1674, 1513, 1087, 631, 317, -826, -922, -1265, -1716, -1826, -1881, -2192, -2488, -2810, -2824, -2867, -3331, -4502, -4596, -5116, -5379, -6051, -6545, -6639, -6831, -7048, -7501, -7537, -8500, -8879, -9077, -9402, -9694, -9964, -10295, -11299, -12285, -12522, -12635, -12818, -13497, -13784, -14059, -14276, -14352, -14566, -15181, -15374, -15628, -15858}; cout << tripleIndexSLM(nums);	0	0	✓
✓	vector<int> nums {16166, 16125, 16094, 16061, 16054, 15970, 15932, 15918, 15883, 15701, 15685, 15631, 15579, 15559, 15548, 15513, 15512, 15400, 15322, 15307, 15027, 14973, 14843, 14698, 14671, 14653, 14579, 14546, 14325, 14289, 14099, 14051, 13949, 13938, 13903, 13594, 13593, 13580, 13528, 13423, 13174, 13088, 13010, 13007, 12994, 12961, 12961, 12935, 12878, 12846, 12768, 12724, 12722, 12660, 12650, 12578, 12498, 12494, 12434, 12411, 12296, 12108, 11844, 11840, 11836, 11815, 11801, 11725, 11654, 11465, 11436, 11390, 11253, 11236, 11169, 11148, 11103, 11088, 11056, 11001, 10913, 10867, 10764, 10745, 10739, 10633, 10606, 10586, 10556, 10504, 10485, 10445, 10439, 10410, 10281, 10254, 10219, 10126, 10098, 9973, 9921, 9827, 9268, 9257, 9227, 9122, 9058, 9041, 9021, 8989, 8871, 8796, 8717, 8553, 8532, 8529, 8484, 8475, 8386, 8343, 8256, 8255, 8235, 8214, 8210, 8150, 8075, 7938, 7775, 7707, 7658, 7637, 7512, 7412, 7397, 7326, 7234, 7167, 7155, 7152, 7106, 7101, 7067, 6917, 6898, 6865, 6748, 6586, 6400, 6203, 6132, 6093, 5952, 5951, 5896, 5854, 5852, 5804, 5788, 5724, 5696, 5646, 5579, 5387, 5346, 5335, 5301, 5124, 5079, 5037, 4860, 4789, 4696, 4645, 4634, 4633, 4532, 4510, 4483, 4339, 4335, 4290, 4265, 4203, 3964, 3829, 3795, 3768, 3657, 3634, 3629, 3577, 3576, 3557, 3301, 3275, 3171, 3016, 2999, 2940, 2847, 2782, 2613, 2452, 2326, 2285, 2276, 2168, 2006, 1997, 1956, 1955, 1943, 1906, 1719, 1665, 1612, 1566, 1376, 1306, 1171, 1061, 1039, 978, 883, 859, 852, 819, 804, 803, 576, 566, 407, 242, 233, 229, 197, 192, 112, 37, -4, -96, -97, -110, -114, -137, -141, -216, -290, -311, -332, -378, -432, -690, -691, -746, -766, -820, -848, -853, -893, -915, -1029, -1129, -1131, -1509, -1602, -1647, -1689, -1710, -1742, -1807, -1847, -1853, -1911, -1977, -2017, -2028, -2078, -2171, -2186, -2204, -2227, -2284, -2374, -2474, -2493, -2590, -2641, -2768, -2823, -2895, -2970, -3033, -3035, -3118, -3203, -3371, -3417, -3505, -3563, -3734, -3749, -3779, -3823, -4030, -4189, -4194, -4272, -4317, -4394, -4443, -4495, -4647, -4650, -4654, -4704, -4737, -4742, -4880, -4887, -4913, -4979, -4994, -5067, -5101, -5313, -5361, -5375, -5393, -5569, -5682, -5708, -5737, -5810, -5813, -5814, -5895, -5935, -5946, -5956, -6028, -6175, -6178, -6262, -6271, -6329, -6349, -6373, -6405, -6442, -6479, -6623, -6624, -6671, -6753, -6817, -6821, -6853, -6861, -6892, -6954, -6964, -6979, -7056, -7066, -7096, -7261, -7289, -7328, -7331, -7345, -7419, -7486, -7511, -7661, -7734, -7752, -7770, -7820, -7847, -7893, -7956, -7956, -8555, -8567, -8765, -8834, -8943, -8977, -8996, -9092, -9122, -9141, -9174, -9466, -9588, -9597, -9687, -9802, -9807, -9832, -9863, -10015, -10142, -10155, -10214, -10259, -10264, -10544, -10581, -10672, -10708, -10735, -10741, -10810, -10899, -10902, -10932, -10994, -10996, -11008, -11008, -11130, -11178, -11363, -11380, -11720, -11770, -11773, -11792, -11975, -11986, -12004, -12011, -12153, -12203, -12246, -12272, -12318, -12343, -12345, -12370, -12529, -12615, -12798, -12964, -12975, -12977, -12977, -12992, -13039, -13047, -13108, -13202, -13210, -13253, -13333, -13336, -13552, -13553, -13569, -13592, -13631, -13707, -13733, -13981, -14064, -14079, -14101, -14289, -14332, -14475, -14503, -14527, -14667, -14740, -14817, -14820, -14892, -15011, -15058, -15079, -15108, -15169, -15217, -15271, -15334, -15357, -15515, -15589, -15594, -15681, -15804, -15962, -15997, -16056, -16214, -16256, -16317}; cout << tripleIndexSLM(nums);	0	0	✓

	Test	Expected	Got	
✓	vector<int> nums {16376, 16371, 16370, 16370, 16322, 16320, 16316, 16314, 16307, 16285, 16277, 16271, 16270, 16269, 16268, 16256, 16238, 16230, 16224, 16217, 16198, 16196, 16186, 16184, 16182, 16182, 16173, 16171, 16164, 16134, 16132, 16118, 16091, 16078, 16074, 16049, 16038, 16022, 16011, 16010, 16008, 16000, 15998, 15994, 15989, 15987, 15982, 15968, 15967, 15956, 15948, 15946, 15939, 15933, 15929, 15923, 15919, 15919, 15918, 15913, 15907, 15900, 15897, 15896, 15895, 15893, 15892, 15886, 15879, 15871, 15870, 15863, 15856, 15853, 15850, 15843, 15831, 15826, 15819, 15800, 15780, 15759, 15757, 15753, 15748, 15745, 15741, 15741, 15731, 15723, 15719, 15692, 15672, 15667, 15667, 15666, 15659, 15658, 15654, 15645, 15643, 15619, 15617, 15616, 15616, 15613, 15608, 15600, 15597, 15597, 15590, 15583, 15561, 15549, 15546, 15545, 15523, 15521, 15516, 15513, 15510, 15508, 15507, 15505, 15462, 15456, 15453, 15447, 15439, 15438, 15419, 15409, 15406, 15400, 15388, 15378, 15376, 15338, 15328, 15319, 15316, 15306, 15291, 15289, 15288, 15286, 15278, 15274, 15272, 15268, 15261, 15257, 15249, 15231, 15231, 15227, 15226, 15217, 15214, 15213, 15211, 15208, 15204, 15201, 15199, 15198, 15186, 15177, 15152, 15151, 15143, 15140, 15131, 15123, 15120, 15117, 15114, 15114, 15108, 15107, 15101, 15100, 15096, 15091, 15089, 15084, 15081, 15075, 15072, 15067, 15049, 15031, 15021, 15014, 15013, 15011, 14998, 14986, 14982, 14979, 14968, 14967, 14962, 14954, 14944, 14934, 14923, 14923, 14915, 14903, 14894, 14884, 14855, 14852, 14841, 14835, 14818, 14814, 14808, 14807, 14803, 14802, 14787, 14784, 14763, 14754, 14751, 14747, 14745, 14732, 14731, 14731, 14727, 14726, 14713, 14712, 14695, 14682, 14682, 14681, 14679, 14662, 14658, 14654, 14639, 14638, 14614, 14611, 14605, 14602, 14602, 14596, 14579, 14578, 14571, 14564, 14558, 14555, 14554, 14537, 14528, 14525, 14521, 14518, 14518, 14509, 14499, 14494, 14472, 14465, 14463, 14461, 14460, 14453, 14452, 14451, 14443, 14442, 14439, 14436, 14435, 14427, 14416, 14416, 14414, 14410, 14399, 14384, 14383, 14379, 14377, 14375, 14374, 14367, 14342, 14335, 14330, 14320, 14310, 14309, 14299, 14297, 14292, 14282, 14278, 14245, 14233, 14228, 14227, 14221, 14216, 14214, 14196, 14178, 14168, 14156, 14155, 14150, 14147, 14146, 14138, 14135, 14117, 14109, 14104, 14077, 14058, 14058, 14031, 14027, 14016, 14008, 14008, 14008, 14006, 13998, 13979, 13979, 13975, 13971, 13969, 13967, 13962, 13954, 13944, 13930, 13923, 13911, 13909, 13893, 13880, 13878, 13875, 13872, 13868, 13864, 13858, 13848, 13839, 13819, 13816, 13806, 13794, 13791, 13780, 13761, 13753, 13746, 13738, 13728, 13726, 13718, 13717, 13703, 13698, 13687, 13682, 13678, 13677, 13675, 13674, 13670, 13662, 13632, 13623, 13619, 13613, 13607, 13603, 13591, 13585, 13574, 13561, 13552, 13551, 13549, 13545, 13535, 13529, 13512, 13511, 13501, 13500, 13498, 13498, 13472, 13469, 13459, 13449, 13446, 13444, 13442, 13438, 13432, 13429, 13424, 13422, 13418, 13411, 13407, 13407, 13406, 13402, 13386, 13382, 13373, 13373, 13364, 13362, 13357, 13357, 13352, 13352, 13347, 13346, 13338, 13338, 13333, 13332, 13311, 13293, 13285, 13283, 13270, 13268, 13240, 13237, 13235, 13235, 13227, 13215, 13194, 13194, 13187, 13183, 13179, 13168, 13162, 13162, 13154, 13143, 13140, 13136, 13129, 13121, 13108, 13104, 13095, 13092, 13091, 13087, 13078, 13053, 13052, 13051, 13048, 13046, 13035, 13033, 13026, 13024, 13023, 13022, 13016, 13016, 13004, 13004, 12994, 12975, 12973, 12969, 12967, 12965, 12965, 12954, 12951, 12946, 12944, 12941, 12938, 12934, 12927, 12926, 12926, 12902, 12900, 12887, 12877, 12859, 12846, 12835, 12833, 12822, 12788, 12780, 12769, 12766, 12766, 12760, 12747, 12745, 12741, 12739, 12737, 12734, 12729, 12727, 12723, 12720, 12714, 12707, 12706, 12677, 12674, 12667, 12656, 12647, 12644, 12628, 12628, 12627, 12617, 12614, 12613, 12601, 12599, 12592, 12588, 12583, 12583, 12580, 12579, 12574, 12566, 12560, 12532, 12532, 12523, 12520, 12491, 12485, 12484, 12481, 12424, 12423, 12400, 12396, 12385, ...snip... 2, -13264, -13287, -13293, -13295, -13314, -13321, -13321, -13328, -13330, -13330, -13333, -13334, -13336, -13342, -13349, -13352, -13365, -13373, -13373, -13374, -13377, -13383, -13409, -13413, -13413, -13414, -13427, -13428, -13434, -13440, -13440, -13446, -13459, -13459, -13460, -13466, -13472, -13481, -13481, -13481, -13482, -13493, -13515, -13521, -13526, -13527, -13535, -13539, -13539, -13540, -13553, -13555, -13555, -13555, -13558, -13559, -13569, -13571, -13579, -13597, -13599, -13604, -13608, -13623, -13631, -13644, -13646, -13653, -13663, -13665, -13672, -13689, -13694, -13719, -13720, -13747, -13762, -13771, -13785, -13806, -13808, -13829, -13832, -13834, -13835, -13847, -13856, -13862, -13863, -13864, -13867, -13875, -13879, -13886, -13888, -13893, -13896, -13900, -13900, -13913, -13921, -13926, -13927, -13936, -13936, -13947, -13955, -13959, -13977, -13989, -13990, -13998, -14005, -14014, -14014, -14018, -14018, -14019, -14019, -14022, -14022, -14040, -14041, -14045, -14056, -14081, -14083, -14090, -14090, -14097, -14097, -14100, -14102, -14104, -14116, -14123, -14128, -14135, -14137, -14138, -14142, -14144, -14148, -14149, -14150, -14159, -14160, -14161, -14173, -14178, -14189, -14193, -14197, -14201, -14204, -14205, -14211, -14212, -14226, -14230, -14236, -14239, -14240, -14240, -14249, -14268, -14276, -14279, -14286, -14291, -14293, -14299, -14316, -14324, -14329, -14348, -14360, -14365, -14366, -14379, -14417, -14423, -14424, -14430, -14431, -14433, -14436, -14466, -14477, -14488, -14496, -14503, -14505, -14515, -14517, -14517, -14519, -14523, -14545, -14561, -14568, -14568, -14568, -14569, -14570, -14582, -14587, -14587, -14594, -14599, -14599, -14604, -14605, -14621, -14623, -14626, -14628, -14633, -14635,	0	0	✓

Test	Expected	Got
<pre> -14638, -14644, -14649, -14656, -14662, -14663, -14669, -14670, -14672, -14675, -14680, -14685, -14689, -14717, -14719, -14720, -14723, -14731, -14732, -14735, -14737, -14750, -14752, -14761, -14767, -14779, -14787, -14788, -14797, -14801, -14809, -14810, -14829, -14831, -14834, -14838, -14841, -14863, -14870, -14872, -14882, -14883, -14884, -14886, -14890, -14892, -14899, -14900, -14914, -14918, -14918, -14919, -14922, -14926, -14945, -14946, -14947, -14949, -14957, -14958, -14959, -14970, -14970, -14975, -14976, -14987, -14992, -15009, -15018, -15036, -15037, -15037, -15046, -15047, -15047, -15080, -15084, -15093, -15098, -15106, -15107, -15108, -15108, -15112, -15119, -15123, -15130, -15131, -15132, -15141, -15144, -15148, -15150, -15154, -15165, -15176, -15180, -15181, -15181, -15198, -15204, -15217, -15221, -15225, -15254, -15256, -15258, -15264, -15264, -15273, -15279, -15294, -15300, -15304, -15313, -15325, -15333, -15351, -15352, -15357, -15360, -15360, -15361, -15372, -15390, -15402, -15402, -15409, -15412, -15418, -15419, -15422, -15422, -15435, -15447, -15451, -15451, -15451, -15459, -15459, -15460, -15472, -15479, -15480, -15484, -15489, -15496, -15499, -15506, -15515, -15541, -15544, -15544, -15549, -15551, -15556, -15576, -15588, -15589, -15591, -15592, -15599, -15602, -15603, -15605, -15606, -15614, -15624, -15625, -15626, -15630, -15632, -15636, -15641, -15647, -15652, -15653, -15663, -15675, -15678, -15695, -15706, -15722, -15730, -15733, -15735, -15760, -15761, -15772, -15774, -15775, -15782, -15785, -15787, -15788, -15789, -15790, -15817, -15831, -15841, -15843, -15844, -15872, -15882, -15886, -15893, -15894, -15903, -15911, -15926, -15929, -15933, -15962, -15963, -15970, -15972, -15974, -15993, -15995, -16002, -16016, -16022, -16023, -16026, -16030, -16044, -16046, -16052, -16061, -16099, -16114, -16115, -16124, -16128, -16129, -16131, -16131, -16136, -16155, -16160, -16161, -16163, -16169, -16175, -16176, -16179, -16180, -16182, -16202, -16202, -16202, -16208, -16215, -16218, -16224, -16227, -16244, -16249, -16259, -16259, -16265, -16285, -16292, -16293, -16298, -16300, -16308, -16319, -16327, -16334, -16346, -16347, -16350, -16352, -16364, -16366, -16381}; cout << tripleIndexSLM(nums); </pre>		



	Test	Expected	Got	
✓	<p>vector<int> nums {15851, 2945, 13368, 4635, 2829, -10155, 14417, 12366, -3530, 6611, -9685, 13835, -15258, 2382, 10457, 1250, -3970, -8332, 10854, 14981, -12812, -15315, 7203, 16336, -53, -2618, 3392, -14074, 963, -5298, 1946, 11102, -15923, -2496, 15060, 11198, 12225, 10083, -8791, 10950, -8877, -1297, 14720, -11760, -956, 4081, 5185, 9910, 14549, -4368, -5016, 9493, -2431, 9169, 15192, -13871, -15125, 12024, 12785, 11290, 8470, 4528, 9060, 3373, -10540, -1587, 4678, 285, 5761, -12144, 5092, 5331, 4848, 12490, -553, 5459, 15270, 2116, 4568, -9150, -6330, -6049, 7125, -14939, 589, -13832, 13892, 13036, 5631, -14194, 10423, 16175, 15266, 6311, 5296, 2231, -12311, 15727, -6985, 13248, -1357, -15965, 12715, 2685, -8458, -2131, 4694, -8611, 1781, 1297, 2371, 3725, -8006, 15623, 1236, -10306, -8512, 3689, 9615, -11992, 4336, 2256, 16057, 15055, -13874, 15611, -15066, 261, 9205, -912, 8271, 6448, -1087, 15461, 4517, 7117, 12619, 8160, 9765, 16309, -14694, -4834, -6611, 6815, 11799, -3534, 14222, -13593, -8894, 7406, 9698, -4466, -2140, -13818, 1312, 2945, 9732, 3955, 8813, -13026, -4642, 1558, -2251, 15092, 7441, -10594, 4731, -13110, 7640, 3499, 11373, -29, -14597, -1229, 6204, -11903, -953, 8486, 2090, -15704, 13115, -2739, -10267, -14044, 12292, -1416, -12013, -11826, 3756, -2856, 7726, 14346, -4978, 15786, -8302, -10604, -14835, -1515, 5844, 5501, -9083, 14678, -9130, 6512, 14466, 15898, 4888, -1990, 7216, -2261, 3793, 5769, -1517, -5297, -8129, 4852, -15368, 7278, 15291, -7620, 9772, 9714, 11792, 210, 747, 6771, 5005, -8807, -15845, -7020, 1369, -7700, -14021, 12072, 12551, 15075, 12182, 6298, -12694, -14382, 700, 13707, 10155, -11757, 12082, 9916, -6248, -3578, -7831, -14801, 8767, -15019, -3347, 12169, -13380, -9603, 14303, 10171, 4431, -2678, -16219, 10209, -14158, 4641, -7200, 3946, -5719, -14652, 9374, -16006, -14886, -4782, -14126, 12684, -3361, 13517, 13313, 9052, 6810, 96, -9667, 8708, 9672, 14241, -8361, -3483, -2855, -5101, 13905, 1224, 6912, -2065, -10990, 10693, -12410, -1500, -12195, 3685, 2403, -11364, -3068, -15485, -1777, 2497, 11097, -11144, -11114, 12864, -12826, 12535, 10731, -13300, 9310, -4555, 11134, 10663, 14803, 9037, 11024, -12131, -13016, -6940, 14022, -3503, -2358, 9224, 3197, -7685, -10597, -2978, -10402, 11250, -8657, -643, -5198, 12839, 2875, 7061, -2825, -3598, 9802, -3244, -13773, -10478, -9806, -758, -9080, 16367, 13511, -13557, -3063, 5921, -4274, 3925, 11980, 9782, 309, 9330, -2496, 1456, -7551, 7584, -8511, -12722, 1927, 8752, 9578, -4211, -4769, 6188, -16341, -12435, -13674, -7620, 13339, 6008, 14371, 13514, 9743, -11519, -6500, -14747, -9877, -15702, 13226, -8349, 4408, 2345, 13243, 3275, 1911, 5451, 1014, -1422, -4427, -4808, -1932, 14435, 12768, 16122, -1363, -4220, -6567, -9671, 11312, -13961, 9290, -2232, 8600, 4031, 14650, -4250, -13893, -14812, -5921, -13763, -10972, -15272, 12339, -15185, 12740, 4826, 16074, 13522, -4380, 3867, 6754, -4295, 7518, 6880, 8380, 4470, 13, -9497, -9302, -10788, 8322, -14131, -9366, -11126, 7160, 7215, 2866, -10843, 15498, -12528, 13324, 6372, 12964, -16006, 8676, 13906, 696, -16280, 7903, -945, -1230, -3100, 14575, -4392, -14675, -1939, 1445, -13303, -4505, 10139, -10170, 14816, -7950, -11258, 7712, 324, -11987, -925, -10978, -1466, 371, 14832, -12133, 3494, -12526, -4589, -16268, -4199, 4607, -3267, 11873, -2103, 9416, -10987, -6779, 15006, 10441, -11309, -1981, -14404, -9565, 14057, -10156, 12749, 2794, 13033, 13590, -15488, 3870, -6493, 5159, 1489, 16093, 13197, -10189, -1140, -11946, -2711, 10762, 13836, 12033, -8927, -2058, 10057, 2958, -3338, -181, -10990, -9189, 985, 5183, -11882, -599, -487, 7164, 7465, 8145, 319, -4876, -10939, -14413, 9527, 2421, -11057, -208, 15536, -6019, 10652, 1113, 5828, -16177, 820, 3366, 988, 2630, -298, -3220, 8654, 10602, -2884, -8951, 13129, -1727, 10911, 13409, 1651, 10088, 3966, 1295, 11984, 5729, -7012, 4308, 5734, 14895, -2104, 12499, 10783, 8992, -11552, -10495, 9582, 10987, 1176, -15401, -16075, -2054, -653 ...snip... 551, 8939, -15086, 186, 10335, 7878, 15354, -13871, -1050, -4342, -6942, 4926, 8044, 8272, 11212, 9102, -8292, 11172, -14234, 11070, -7092, 11667, -5929, -11150, -11936, 4997, -1609, -1303, -4373, 13134, 12127, 8241, 7320, 255, 13412, -4073, 5193, -10494, 8542, -5790, 16381, 1126, 8686, -319, 9975, 5495, -5377, -15359, 13853, 2706, -6581, -1149, -6242, -3785, 14113, 6126, 3543, 4960, -922, 1569, -14984, 12561, -3109, -11412, -11550, -1999, -7523, 1023, -4396, 11928, -6414, 11682, -14777, -15767, 5954, 10919, 12694, 13474, 4708, -1763, -6430, 4593, -6884, 6727, 11189, -4662, -5919, 1613, -13851, 5046, -29, 2096, -13380, -7442, -11700, 5635, -15531, 2088, -3898, -1703, 8521, 2481, 62, -9222, -8564, -9796, -11544, 3290, -623, -7430, 4740, 4427, 9909, -15510, -8225, -10639, -7206, 10611, 2972, 5878, 16210, 1840, -13260, -14080, -13824, 4107, 5436, 10336, -13127, 13453, -3097, 12557, 10480, 4212, 5323, 7632, 11411, -250, -10520, 16192, -8912, -13914, 4070, 16113, -420, 9381, -6754, 8937, -10411, -4518, -7972, 9555, -1050, -7051, 7523, 13432, 7307, -15472, 15154, 12010, 8662, 1287, -2810, 12290, 5905, 12042, 32, 141, 10277, -5392, 6472, -61, 14011, -10802, -7406, 6265, -13664, 16201, -1696, 5783, -7826, -13656, -8664, -3893, -7831, 9774, 9080, -9601, 13223, -13146, -2377, 12809, -13964, 1343, -3795, 7970, 4340, 5530, -11834, -2662, 7434, -9694, 1042, 12362, 1145, 14492, 6289, 6217, 15433, 8446, 4654, -5021, -3731, 4229, 11529, 7366, 9634, -13566, 633, -4353, 5018, 15989, 11858, 7717, 7770, -15140, -9885, -13484, -12319, 14413, -12989, -11683, 11018, 3395, -2738, -10564, -11085, -6346, -3862, 7371, -10209, -9362, 6927, 821, 2713, -10728, 11014, 12460, -11686, -10854, -11664, -13857,</p>	1	1	✓

Test	Expected	Got
<pre> -11091, -4060, -885, -15450, -12463, -10361, 4785, 9600, 8479, -12469, -5599, -12224, -14868, -14085, 16130, 7427, -13903, 6852, -8027, -6043, -2326, 1087, -3317, -8715, -15301, -2678, -5721, 6118, -15015, -13155, 8036, -6393, 9000, -805, 12266, -10507, -8415, -11032, -2531, -15730, -6665, -10565, -11289, 3242, 13690, -10764, 10040, -12026, 6587, -12177, 4171, 8577, -14991, 13593, -4822, 9099, -13925, 3330, 11892, 8666, 12391, 15028, -5794, -6562, 16211, -15420, -13576, -3976, 9362, -10074, -12110, -9369, 5041, 3781, -5602, 2307, 5498, 211, -6153, -6995, 3865, -13986, -9117, -11642, -15235, 2981, -12171, 7903, -9727, -343, -3338, 497, 6476, 11911, 9317, 3022, -651, -6140, 1084, 3224, -13717, 618, -7307, -13071, 13205, -1133, -16307, 10930, -10827, 6723, 12104, -4640, 2066, 2747, -10891, -5668, -14122, 10634, -10253, 16089, -14155, -3929, -6988, -12397, -9777, 15037, 14465, 16191, -14888, -12090, -14786, 3005, -12426, -2597, -7312, 6165, -7446, -6838, 7676, 15137, 14835, -2326, 7985, 9863, -2321, -8789, -16375, 14827, -1212, 13425, 12972, 5730, -770, -13021, -6858, 1577, 8122, -2562, 442, 3223, 5314, 10544, 12167, 12880, 1657, -9566, -1622, -7748, -11526, -10219, -332, -6217, -9732, -4240, -12867, -4982, -7687, -12456, 5288, -1766, -10097, -10750, 937, -9120, -3546, -15803, 2565, 1305, -4909, -1292, -12263, 9151, -1412, -8086, 4766, 10423, 5693, -9114, -15573, -3380, -14069, 9242, 4603, -15908, 8599, -8066, -15119, -15919, -7307, 12522, -39, -15917, 12670, 15332, -7695, -5187, 14195, -15817, 7336, 4045, 5859, -6075, -6235, -10081, 6990, 10653, -2549, -1188, -1654, 10960, -12036, 14383, 7782, -8831, 15196, 16120, -12146, 10940, 727, -13243, 10670, 6025, -7643, 4734, 10970, 6080, -10100, -15521, -4050, 798, 6823, -12885, -14415, -8039, 103, 8991, 12159, -1039, 5207, 6821, -216, -1656, 10276, -15669, -16301, -9101, 4700, 14451, -1369, 13009, -1674, -10804, -2325, -9125, -12269, -12927, 15723, 7997, -6525, 1584, -11362, -219, 13707, -13795, -761, 2713, -14747, 7976, 857, 4749, -9207, -9476, -13858, 53, -15957, -14454, 12313, 15640, -7547, -5194, 1963, -7834, 3798, -13233, 8563, 7307, 4651, 14661, 13817, 3076, -291, -1258, 11533, -2959, -11557, -2934, 2219, 4332, -9815, 3232, -2270, -4966, -4183, -2872, -12171, -1409, 2714}; cout << tripleIndexSLM(nums); </pre>		



	Test	Expected	Got	
✓	<p>vector<int> nums {770, -8297, 11102, -1497, 13153, 8994, 7424, -15945, 5007, 14655, 9461, 50, 13644, 7100, 2210, 3323, -9279, -6867, 3103, 8282, 4198, 8763, -12741, -12431, -221, -12993, 9664, 1137, -8778, 10256, -7176, -14579, -7596, -6120, 8303, -5841, 3475, -5970, -3589, -5142, -13433, -10336, 14586, 12465, 12331, 4110, -8485, -13626, -11517, 6015, 12000, 1268, 3584, -3794, 11297, -4223, 9124, 15269, 2261, 2655, 4198, -13171, 8909, -8286, 2904, 16069, 10561, 8759, -1529, 9905, -12337, 9426, -9309, -7979, 12993, 15656, -4256, 11938, -12145, -6305, 6683, 5519, 15524, 15810, -12479, 13420, -4251, 6330, 237, 755, -10384, -5383, -9392, 12193, 4129, 11769, -12458, 4036, -7845, -3419, 12422, -10796, -4987, -5546, -5589, 518, 10962, -1024, 4017, -16097, 8390, 14766, -13231, 11654, -5386, -13781, -2575, 9886, 2063, 6914, -14523, 12882, 1049, -4439, 1339, 7410, -5891, 6906, 5014, -13507, 6213, 3042, 15576, -3393, -3548, -8611, 12646, 11357, -4682, 10146, 487, 5422, 1672, 15514, 6381, -3700, 9051, 3479, -2089, 1643, -9691, -5565, 15094, 7402, 13657, -15913, 10478, -7128, -8600, 13197, -11937, 9561, -11896, 855, -315, -487, -332, -14639, -99, 5375, -1036, -8588, 2762, -906, 11486, 3853, -15562, 14280, 861, -12434, 7868, 5756, 15793, 7780, 13382, -9758, -14921, -2839, 13618, -10004, 5483, 10394, 5627, 5695, -15189, -11669, 10773, -886, 9278, 3481, 14104, 13591, 2580, 8928, -15583, -13152, 13007, -3089, 8821, 6755, -9085, 15129, 2152, 7544, -326, -3251, 8062, -15261, -15428, -952, -4913, -11546, 9997, -12535, 688, -9703, -8906, -10619, -14705, -2372, -5944, -16227, -12769, -13877, 159, -9395, 6070, -8365, 11108, -11738, 8691, -2171, 13397, -8194, 9891, -6648, -8887, -14106, -795, -1132, -3602, 12380, -6515, 11709, 8278, 8130, -13113, -3828, 15026, -12017, 12285, -1708, -14755, 4549, -12550, -11197, -996, -11741, -14001, 4664, 7588, 13467, -16139, -3756, -15268, 12678, 16181, 16084, 3429, 8983, -7995, -7522, -14632, -12915, 3380, -14055, 590, 14253, 15751, 8002, 12454, -6288, 5669, 5373, 11010, -10503, -10256, 6568, -13037, 13769, -4939, -14602, 1175, -13770, 4806, -4613, -1505, 1817, -5147, -1405, 14102, 10696, -3228, 8360, -4865, -3004, 8075, 2212, -15874, -10317, 12858, -4893, -14694, 8976, -817, -8611, -1855, 2593, 6986, 8994, -2036, 6313, -5456, -7370, -12475, -6688, 16045, 6048, -2265, 7200, 6780, -4162, -2460, 10465, -3828, 8411, -15635, 10471, 12379, -11261, -14227, -6982, 7206, 9750, 14358, -3753, 15301, -4580, 1441, -14043, -8367, 15832, 13712, -3798, 7, -12128, -8009, -385, -15, -9108, 4346, 4097, 3267, -1748, -9384, 4170, 10755, 4703, -9431, -11296, 9836, 5857, -3421, -6229, 149, -9468, -2909, 9133, -306, -7965, 7083, -10771, -438, -1293, -3546, 6580, 15748, -805, 4084, 12028, 9691, -4474, -11998, -6618, -10113, -2698, -5639, -3093, 657, -6053, -965, -15979, 15494, -16156, 4521, 3439, -1843, -8235, 8136, -4230, -14927, -10406, -1819, 5729, -6528, 12186, -12713, -12113, 2029, -9840, 10244, 13888, 5043, -15338, -2777, 1525, -6277, -14525, 6736, 2988, -3328, -9737, 8364, 8816, 369, -14924, -5498, 9964, 12823, -13136, 4366, -4570, -7113, -2520, -12455, 5226, 6714, -2826, 7850, 14467, 6155, -9114, -6160, -2254, -16191, 5958, -4465, 15064, -2178, -10948, -10141, 13174, 1942, -7587, -2302, 2275, -10252, 7418, 13890, 10508, -4678, -579, -11268, 7449, 552, -6667, -15089, -530, 12223, 10287, -5209, -10343, -1370, -250, 703, -1863, 14412, -6596, -7452, 13366, 6735, 2569, 1084, -5329, -8035, -16206, -7736, -3276, -2343, -15778, 8678, 11446, 256, 4902, 10266, -8773, 6988, -11413, -3059, 9430, -11944, -5524, -3626, 6231, 4230, 16151, 8057, -6721, 6558, -3296, 11181, 1399, 13751, -11183, -3613, -10559, 12891, -2015, -2173, 14318, -5914, 13703, -12679, -4398, -10961, -3824, 240, -2699, -150, -4304, -5539, -245, 15693, 8405, 8640, -14042, 3698, 7746, -9945, 3371, 6141, -15333, -10637, 9730, 894, 9916, -5794, 15890, -15079, -12337, 701, 6509, 3300, -1506, -12029, -12217, -6256, 16293, 1543, 3501, -11700, -4628, 10759, 9844, -14874, -2924, ...snip... 67, 12326, 1825, -6696, -660, 16247, 13104, -4633, -3635, 10475, -8622, 1093, -859, -8503, -1852, -15603, -12491, 3745, -2586, 7149, -3837, -13265, -9326, -1495, -5680, -7295, -9300, -14877, -14389, 5724, 11258, 11417, -10748, -14115, 4150, -11537, 5923, -7844, -13726, -11345, -10150, -8299, -6820, 12302, -3944, -4342, 3506, 5425, 0, -1885, -4937, 6895, -15463, -2152, 14807, -8231, -11961, 5520, -2521, 9847, -12294, -11765, 1600, -10283, 37, 14589, 7758, 3902, 10354, -4933, 13412, 4936, 15028, 1819, -4845, -8175, -15076, 5591, 13882, 9927, 945, 13875, 1055, -6546, 9722, 5088, -6977, 477, 7421, -12513, -8861, -2294, 1390, 3626, -14732, 7048, 544, 15823, 6544, 590, 6847, -15010, -6691, -4422, -3391, 12741, -8546, -1834, -4966, 7162, 14340, -14631, 3375, -12647, -14778, -15, 13198, -4401, 4852, -9153, -1828, 11382, -9080, -2525, -13170, 15745, 13374, -16197, 8413, 10395, 11813, -14426, 15219, 8971, -10378, 7182, -405, 1499, -6363, 4837, 15736, 7737, -9033, 14446, -3228, -8931, 339, 7424, -5187, 10796, -229, -8441, 13515, -8739, 701, 5707, -4668, 9946, 6980, 1141, -15037, -8535, -13259, 11983, 15351, 4491, 1495, -15029, -104, 15780, -2094, -1337, -3552, 3421, -3589, 12943, 12599, 4359, -14481, 10019, 12783, 15297, -6177, -4549, -15183, 10139, 9002, -503, 5496, -6917, -13519, 6078, -964, -9920, 10215, 15183, 12722, -3930, -3673, 2169, 15410, 10089, -14901, 12547, 10488, 11822, -13958, -8125, -6768, 11678, -4415, 15331, -12916, -4780, 4069, -7707, -16264, -7687, -11016, 2021, -11248, -6483, 6059, 8094, -2384, 9641, 14691, 9392, 5643, -3037, 8079, -2143, 7712, 9586, -13768, -5599, -7062, 6515, 14553, -8046, -11433, 2603, 4466, 11463, -5636, -9042, 15832, 1548, -7161, -1729, -3249, -15769, 7669,</p>	1	1	✓

Test	Expected	Got
<pre> -4424, -10150, 13598, 2437, -6867, 15570, -10631, -14740, 9164, 6198, 6345, 13130, 11783, -15806, 3753, 2781, 15350, 6595, -14716, -7820, 4598, 3497, -168, 5799, -1129, 15454, 6841, 3910, -4683, 7510, -4072, -13425, 5092, -5912, -6843, 11684, -13038, 12892, -4045, 1084, -9369, -6075, -6660, 11672, -13456, 12049, 15140, -6471, -10627, -13858, 6434, 6864, -15289, 1423, -4836, -14602, 4054, -9697, -11277, -14819, -11358, 13878, 1919, 6470, -16135, 10490, 12572, -11, 9888, -8664, -15003, -2846, -13635, 8291, -12993, -7438, 7156, -10804, -16358, 9829, -559, 10962, -11405, 11857, 13414, -8372, -7267, -11697, 13109, -5691, -9790, 12404, -11440, -6890, 13093, 10115, 13502, -965, 3497, -16032, -12511, -14241, -8809, 8648, -2666, -10951, 12444, -12992, -11811, 16283, 8721, -9141, 6842, 8776, -7041, -4384, 2458, -15097, 1123, -3045, -11398, -15883, -14290, 6707, 10501, 5407, -11888, -1364, 508, -13744, -7178, -10546, 14299, 15656, -2990, -1963, 13242, 8046, -6630, 2666, 13169, 14759, 6547, -302, -5633, 6632, 12111, -11418, -211, 9448, 9460, -13373, 12910, 15705, 12096, -3262, 9482, -2695, -2742, -1237, 3963, -284, 4363, -14043, -6633, -13140, -1764, -13839, 12528, 7083, 14273, -9128, 6905, -10984, 4903, -6046, -918, -167, 2794, -2359, -7067, 12568, -12582, -9043, -6695, 15001, 4863, -12407, 11175, -1717, -6949, -2714, 6147, 2461, -6674, -15025, -3299, -13550, 3968, -13308, -4863, -5480, -7270, 10483, -2372, -6378, 12118, -10397, 14017, -15292, -15784, 6131, 7307, -8257, 16285, 3651, -10365, 7261, -5260, -6415, 13603, -10307, 6857, 6599, 10844, -8669, 10153, 7218, 3125, -12157, 3843, 6134, 15286, 12508, 7084, -1503, -14076, 2248, -15706, -4547, -11827, -8624, 5282, 6602, 7910, -5372, -5949, 11244, -15053, -14004, 9381, 5472, -12856, 3065, -14946, -844, 10573, -3041, 3908, -15688, -3417, 14631, 815, 11056, -15147, 5920, 1536, -565, 10263, 5974, -10847, 13145, -9642, -14554, 13519, 824, -2368, -12016, -4138, -6897, -16205, 11278, 7836, -9537, 5596, -10972, -14367, 5783, -13408, -14535, -13883, -11832, -1094, 9509, 11147, -6376, 1062, -13813, 15516, 1498, 15934, 13289, 9423, 4381, 12580, -915, -13849, -220, -15521, -11253, 11870, -13034, 12233, 13132, 11617, -9481, -15960, 13929, 14363, -15401, 7195, -10472, -1994, 1609}; cout << tripleIndexSLM(nums); </pre>		



	Test	Expected	Got	
✓	<p>vector<int> nums {-1882, -1121, 4375, -13965, -1466, -12918, 2100, -13887, -15447, 8375, -10766, 2814, -50, -4365, -9099, 14348, -1508, 12769, -9368, -13250, -11858, 6394, -8722, 6516, -2322, -7981, -15140, -1657, 10085, 12190, -16091, 12007, -9263, 6900, -11626, 14466, -9096, 15058, 7875, 7400, -4163, -4639, 13377, -3923, -10511, 13050, 3605, 7183, 3435, 15337, -5267, 9177, -549, 7537, -12402, -1656, 8667, -16239, 15886, 5803, 16327, -1786, 1859, -6256, 15055, -6895, 14587, -14052, 10570, -1414, 1901, 7675, 9549, -6955, -15760, 16122, 2941, 6404, 10987, -13648, 7201, -15875, 15501, 6244, 12924, -5426, 16318, -11733, 1625, 2526, 14293, -1455, 11239, -11915, 395, -8198, -14905, -5762, 13743, -14583, 7524, -2420, 9244, -14527, 5507, 4636, -1303, 9898, 11035, 6170, -4641, 13111, -1946, -4698, -3536, -8283, -4552, 9495, -7579, 10150, 14224, 9024, 7833, -193, -1172, 10739, -795, -11921, -3870, -5180, 7707, -11759, -1683, 3343, -1353, 9987, 13701, 13885, 15378, 11400, -14478, 6978, 13785, -10005, 14433, 7274, 5054, 870, 13240, -7082, 3949, 4673, 3239, 10049, -15678, -11202, -15873, -6263, -5575, 2357, -3000, -7318, 11174, -8708, 526, 4108, 13606, -8311, 15995, -14912, -14251, 10587, -2517, 390, 1473, -11498, 13883, -8236, 3231, -7827, 2467, 13458, -15683, 15940, 8390, -7063, -10056, -1002, 5960, -7363, -8593, -5995, -13280, -1032, -1097, 12945, -18, -14966, 8529, 10011, 11662, -5842, -14823, -13846, -12051, 7662, 1151, -13058, 4123, -663, 11323, 11138, -1374, 5588, 7229, -5129, -11497, -8897, -1821, -12913, 5870, -5607, 4719, -6953, 14482, 5367, 4024, -5223, 16310, -6255, 12576, -14335, -15387, -755, 16325, 3838, -2719, 14209, -772, 1515, 10282, -4429, 9723, -6289, 5469, -989, -8235, -4190, -14657, -14261, 8108, -3215, -8038, 15653, -7391, 1859, 15816, -15890, -13040, -1827, 4621, 9258, -1413, 8322, 1706, 4584, 9155, -12961, 1326, 2375, -8674, -11993, 9360, -7879, 4421, -4357, -4997, -12409, -2707, 12684, -7447, 4117, 11641, -4989, 6131, 4456, -11922, -4831, 9191, -13723, -729, 7714, -13412, 7515, 14154, 8936, 16130, -5052, -15238, 11733, 9725, -15800, 15054, 5625, -4236, 13955, -5758, -4115, 5797, -5112, -5907, 11774, -11589, -11482, -6477, 2958, -7006, 4074, 572, -14408, -2278, 10724, -12955, -3277, -15094, -5842, -14987, -10887, -1513, 1195, -10158, 11560, 374, 1421, -13835, -12376, 14078, 1050, 3611, -5106, 11906, 8623, -5571, -2752, 1373, -7163, -3133, -2214, -3506, -10829, 15989, -9135, 1927, 4084, -12494, -7209, -9472, 4523, 9956, -11650, 2321, 1956, -1613, 7199, -15040, -5603, -6535, 3453, -4776, 4997, -15754, 14630, -11589, 13443, -12986, 11162, 11768, -14022, -7921, 10927, -3936, -3124, -3329, 1076, -5802, -8164, -2058, 12245, -4570, 5337, 383, -14274, 280, -10824, 9989, 8068, 8964, -1280, 3908, -3392, -14169, -7604, 11773, 253, 9719, -9076, 2542, 2388, -15436, 15694, 7937, -2987, 4928, -4292, 12484, 11433, -2896, 12773, 8473, -5386, -7389, -6976, 7983, -1789, 15186, 877, -15696, 12317, -12183, 14143, -6340, 14009, -15594, 8696, -9193, -16038, -12577, -14893, 15390, -9547, 440, 11976, 512, -2285, 4565, -16117, 11434, -10504, -14915, -12074, 2051, -766, 8864, -6296, -4646, 15123, -11030, -9906, 4892, 6835, 1816, 4577, 13643, 9391, 6858, 14352, 15403, -13306, -2022, 4191, 9151, 6, -6355, -16336, -15757, 10675, 10247, -10687, 16283, 6248, 15524, 15479, -12772, 8641, 6468, -2342, -151, -1865, 12038, -11296, -15694, 13, 1208, -3602, -7080, 7148, -11794, 15900, 8611, -14646, -12207, -12231, 9399, -4720, 14138, -14321, -828, -10501, 5504, 4071, -6530, 6722, -626, 10320, -1843, -5894, 14264, -7907, 2501, 12343, -4401, 9133, -8232, -11019, 2802, 12911, 842, 4022, 11944, 420, -15386, 11116, 2604, 3431, -14608, 11160, -1784, -12053, 12634, 7604, 7653, -15611, -15814, 6607, -3934, -1046, -3716, 13943, -3421, -6087, -6444, -3109, -660, 11580, -8969, -7631, -8338, 8811, -15093, 1878, 15136, 1073, 15964, -14412, -9134, -15373, 13904, -10647, -2113, 6479, -15203, -3930, 2273, 14497, 6073, -6060, -15863, -11572, 1427, -15874, -15698, ...snip... 592, -2134, 3574, -12128, -1754, -5512, -8055, -16337, 15325, 6241, -2595, 7997, -13097, 14668, -811, 7536, 10285, -2118, -9866, 9225, 4653, -14700, 13211, -1593, 15278, -15322, -8727, -8696, 160, -12693, 10502, -8398, -2841, 11403, -2298, -4015, 13162, -3678, 8998, 514, 9062, -829, 4541, 8847, 10249, 14909, -913, -5797, -4548, 2588, -12636, 12673, -13120, -4109, 9784, -6216, -5508, -4135, -13195, -4230, 7957, 15538, 15166, -7323, -15936, 7088, 652, 13537, -9644, 8473, 8735, -4349, -5980, 10826, 11117, 8369, 13835, 14588, 16351, 12807, -6840, 5062, 16188, 10490, 13452, -12317, 14949, 8507, 9682, 10294, -444, -1186, 367, -14965, -13569, 1852, -6422, -9184, 2929, 11670, -8179, -15772, 8410, -8965, 10812, -6914, 5845, 15851, 10758, 15377, -11381, -3524, 15610, 9662, 6183, -7235, -8772, -13845, -6774, -6630, -12712, -5309, 6463, 12287, 15573, -13954, -10575, -4224, 5646, -1821, 4251, -583, -1347, -11509, -14655, -2639, 2777, -6937, -13346, 12933, -1246, -507, 10205, -1854, 10490, -15090, -5557, -12526, 8310, -2668, 269, -13797, -14155, 11018, -6322, -12380, 10864, -8940, -16196, 5369, 1235, -3838, 13556, -9071, 9355, 1174, 7994, 14063, 12577, -10738, -6546, -13626, -7034, -4770, 4411, -6048, -7953, -5442, -5486, -5056, 6370, 7780, -13737, 13575, -623, -12909, -10662, 12436, -6296, -14124, -12408, -15249, -4894, 14212, 16227, 5480, 7072, 15502, 6575, -3461, -14404, -10118, 9838, -6409, -6821, 11431, -14321, 14304, 8321, 980, -3613, -12143, 10026, -15387, -7472, -7515, -9834, 11492, -1975, 15640, -11036, 2375, 11645, 7850, 13169, 6863, 1924, -7644, 5243, 2439, -12770, -685, 51, -4153, -1184, -16184, 4709, -5952, 12808, 10073, -3750, -731, 9547, -4237, 6559, -12320,</p>	1	1	✓

Test	Expected	Got
<pre> -7489, -12950, 5895, 773, -3208, 6156, -12557, -5884, 6257, 9084, 16337, -10718, 9774, 15934, -9757, 9770, -15456, 1166, -10942, -13729, 4169, -522, 10530, -3146, -2655, 14623, 2135, 11586, 10167, -9752, 1817, -1296, 6427, -12875, -12796, -12172, -6660, 6665, -414, -2941, -1394, -1704, 11181, 9283, -1874, 14870, 3664, 2816, 8393, 5140, -11346, -7648, -15878, -6132, 12133, -12967, -2702, 7547, 4564, -14559, 10525, 4782, 10203, 12646, 15747, -11488, 2017, -8576, 15084, 2051, 12316, 9828, -7737, -10467, 14860, -9190, 6732, 9173, -6327, 14193, 1889, -6514, -5249, 1777, -16198, 3635, 6071, 15889, -1979, 8211, -13608, 13226, 3781, 2544, -13459, 9388, -9284, -9646, -4440, -12528, 11407, -11487, 12455, 14605, 16101, 2675, 8695, -1905, 14928, 3491, -12713, 3813, -4817, 15011, 190, 11449, -12077, -15469, 2207, -3514, 6368, 4242, -7061, -59, -11786, -3116, -2728, -5592, -11375, -7180, 8289, -6133, 2350, 12428, 141, -13592, -10576, -6908, 3701, 6282, 7169, -12889, -7936, -10044, 7164, -11524, -5262, -5278, -13685, 8019, -9051, -1440, -14836, -15714, 917, -9126, -6, -14206, -8925, -1212, -10039, -2898, 12050, 4955, 13073, 6508, -11178, -8613, -15889, 15477, 7822, -269, 5801, -502, -3889, -16266, 12688, -2496, -2141, -301, 6699, -868, 8785, -13625, -15727, 3551, -16033, 7761, 14871, -15325, -4522, 13671, -3285, 5415, 8389, -8473, 8320, -13592, 5256, -9776, -4756, -5519, -6838, 4786, 5518, -781, 841, -10909, 8115, -4328, -6823, -14604, 16378, -1387, 7212, 15331, 9527, -12801, 10960, 16018, 13718, -6433, -14215, -9736, -383, -11768, -4223, -873, -9277, -14808, -8787, 1966, 6637, -13154, -5710, -8529, 6136, -4981, -11784, 2803, 11309, 3116, -5987, -14419, -12222, -921, -16195, 2589, -10042, -7517, -16143, -9722, 15295, -2851, 16149, 3681, 10897, 15352, 2383, -15789, 801, -12203, 76, 5725, -9157, -15196, 9933, 4563, 12593, 13050, -13725, 14479, -3899, 8227, 10614, 10470, -3255, 10553, 14988, -10789, 7173, 12149, -1848, 12691, 10713, -2490, -9807, -9207, -1244, -6240, -12234, 10755, -14108, -3433, 238, 6213, 7454, 14380, 2980, -252, 2176, -1433, 3602, 6557, -9403, 422, -15379, 13959, -6823, -5068, -6110, 9094, 445, 7197, 15200, 10786, -3794, 14145, -11306, 12376, -14335, -4696, 6718, -11889, -3246, -1515, 6170, 974, 6493, 4318, -1219, 7467, -9940}; cout << tripleIndexSLM(nums); </pre>		



Test	Expected	Got
<pre> 16277, 16278, 16278, 16278, 16279, 16279, 16279, 16279, 16280, 16281, 16281, 16281, 16282, 16282, 16282, 16283, 16283, 16284, 16284, 16285, 16285, 16285, 16285, 16285, 16285, 16285, 16286, 16286, 16286, 16286, 16287, 16288, 16288, 16288, 16289, 16290, 16291, 16293, 16294, 16294, 16294, 16295, 16295, 16295, 16295, 16296, 16296, 16296, 16297, 16297, 16297, 16298, 16298, 16298, 16298, 16298, 16299, 16299, 16299, 16299, 16300, 16300, 16301, 16301, 16301, 16302, 16302, 16302, 16303, 16303, 16303, 16304, 16304, 16304, 16305, 16305, 16305, 16305, 16306, 16306, 16306, 16307, 16307, 16307, 16308, 16309, 16309, 16309, 16309, 16310, 16310, 16311, 16311, 16311, 16311, 16311, 16312, 16312, 16312, 16313, 16314, 16314, 16314, 16314, 16315, 16315, 16315, 16316, 16316, 16316, 16317, 16317, 16318, 16318, 16318, 16318, 16319, 16319, 16319, 16320, 16320, 16320, 16320, 16320, 16321, 16322, 16322, 16322, 16322, 16323, 16323, 16323, 16324, 16324, 16324, 16324, 16325, 16325, 16325, 16326, 16326, 16326, 16326, 16327, 16327, 16327, 16327, 16328, 16328, 16329, 16329, 16329, 16330, 16330, 16331, 16331, 16331, 16331, 16331, 16331, 16332, 16332, 16332, 16333, 16333, 16334, 16334, 16334, 16334, 16334, 16335, 16335, 16335, 16335, 16336, 16336, 16336, 16336, 16337, 16337, 16337, 16338, 16338, 16338, 16339, 16339, 16339, 16339, 16340, 16340, 16340, 16340, 16340, 16340, 16340, 16340, 16340, 16341, 16341, 16342, 16343, 16343, 16343, 16344, 16344, 16344, 16344, 16345, 16345, 16346, 16347, 16347, 16347, 16347, 16347, 16347, 16348, 16348, 16348, 16348, 16348, 16349, 16349, 16350, 16351, 16351, 16352, 16352, 16352, 16352, 16352, 16352, 16353, 16353, 16354, 16354, 16354, 16355, 16355, 16355, 16355, 16356, 16357, 16358, 16358, 16358, 16359, 16359, 16359, 16359, 16360, 16360, 16360, 16361, 16361, 16362, 16362, 16362, 16362, 16362, 16362, 16362, 16363, 16363, 16363, 16363, 16364, 16364, 16364, 16365, 16365, 16365, 16365, 16365, 16365, 16365, 16365, 16366, 16366, 16366, 16367, 16368, 16368, 16368, 16368, 16370, 16370, 16370, 16371, 16371, 16371, 16371, 16372, 16372, 16372, 16373, 16373, 16374, 16375, 16375, 16375, 16375, 16376, 16377, 16377, 16377, 16377, 16377, 16378, 16379, 16379, 16379, 16379, 16380, 16380, 16380, 16381, 16381, 16382, 16382, 16382, 16382, 16382, 16383, 16383, 16383}; cout << tripleIndexSLM(nums); </pre>		

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

