How to perform operations for an array or vector etc in C?

[intro]

Transform

There are a method in the library <algorithm> - transform.

[namespace]

std

[library]

<algorithm>

[syntax]

template <class InputIterator, class OutputIterator, class UnaryOperator>

OutputIterator transform (InputIterator first1, InputIterator last1,OutputIterator result, UnaryOperator op)

where

first1 is an iterator to point the begin of input.

last1 is an iterator to point the end of input.

result is an iterator to point the begin of output.

op is the operation you want to perform.

[Example]

Suppose I want to perform the following statement.

#include <iostream>

#include <vector>

#include <algorithm>

#include <functional>

using namespace std;

void PrintVector(vector<int> vec1)

{

for(int i=0;i<vec1.size();i++)

{

cout<<vec1[i]<<" ";

}

cout<<endl;

}

int main()

{

vector<int> T,X,Y;

X.clear();

Y.clear();

for(int i=0;i<10;i++)

{

X.push\_back(i\*i);

Y.push\_back(50+i\*10);

}

T.resize(X.size());

transform(X.begin(),X.end(),Y.begin(),T.begin(),plus<int>());

cout<<"X="<<endl;

PrintVector(X);

cout<<"Y="<<endl;

PrintVector(Y);

cout<<"T="<<endl;

PrintVector(T);

return 0;

}

[output]

I run my code in ideone, getting the expected result.

X=

0 1 4 9 16 25 36 49 64 81

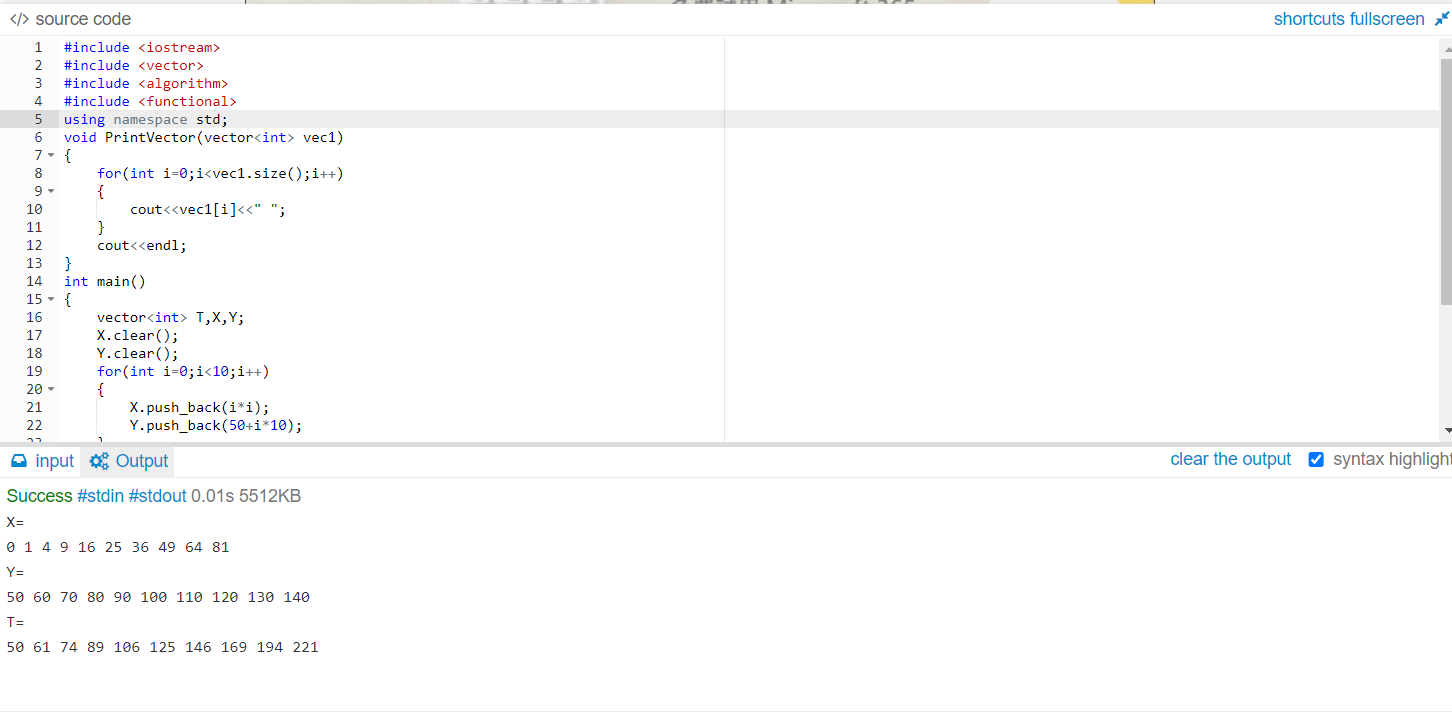
Y=

50 60 70 80 90 100 110 120 130 140

T=

50 61 74 89 106 125 146 169 194 221

[screenshot]



[figure]

The following figure illustrates the arguments of transform used in the above example.

X.begin()

X.end()

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | … | … | 8 | 9 |
| 0 | 1 | 4 | 9 | 25 | … | … | 64 | 81 |

Y.begin()

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | … | … | 8 | 9 |
| 50 | 60 | 70 | 80 | 90 | … | … | 130 | 140 |

T.begin()

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | … | … | 8 | 9 |
| 50 | 61 | 74 | 89 | 106 | … | … | 194 | 221 |

[equivalent code]

Here is the code of implementation of transform.

template <class InputIterator, class OutputIterator, class UnaryOperator>

OutputIterator transform (InputIterator first1, InputIterator last1, OutputIterator result, UnaryOperator op)

{

while (first1 != last1) {

\*result = op(\*first1); // or:\*result=binary\_op(\*first1,\*first2++);

++result; ++first1;

}

return result;

}

[ref]

transform:

<https://cplusplus.com/reference/algorithm/transform/>

ideone:

<https://ideone.com/>