#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

std::vector<unsigned int> readInCoinValues();

int main()

{

std::vector<unsigned int> coinValues; // Array of coin values ascending

int ticketPrice; // M in example

int paidMoney;

coinValues = readInCoinValues();

cout << "ticket price: ";

cin >> ticketPrice;

cout << "money paid: ";

cin >> paidMoney;

if(paidMoney <= ticketPrice)

{

cout << "No exchange money" << endl;

return 1;

}

int diffValue = paidMoney - ticketPrice;

std::vector<unsigned int> coinCount;

for(auto coinValue = coinValues.begin();

coinValue != coinValues.end(); ++coinValue)

{

int countCoins = 0;

while (diffValue >= \*coinValue)

{

diffValue -= \*coinValue;

countCoins++;

}

coinCount.push\_back(countCoins);

}

cout << "the difference " << paidMoney - ticketPrice << " is paid with: " << endl;

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

{

if(coinCount[i] > 0)

cout << coinCount[i] << " coins with value "

<< coinValues[i] << endl;

}

return 0;

}

std::vector<unsigned int> readInCoinValues()

{

// coin values

std::vector<unsigned int> coinValues;

// make sure 1 is in vectore

coinValues.push\_back(1);

// read in coin values (attention: error handling is omitted)

while(true)

{

int coinValue;

cout << "Coin value (<1 to stop): ";

cin >> coinValue;

if(coinValue > 0)

coinValues.push\_back(coinValue);

else

break;

}

sort(coinValues.begin(), coinValues.end(), std::greater<int>());

auto last = std::unique(coinValues.begin(), coinValues.end());

coinValues.erase(last, coinValues.end());

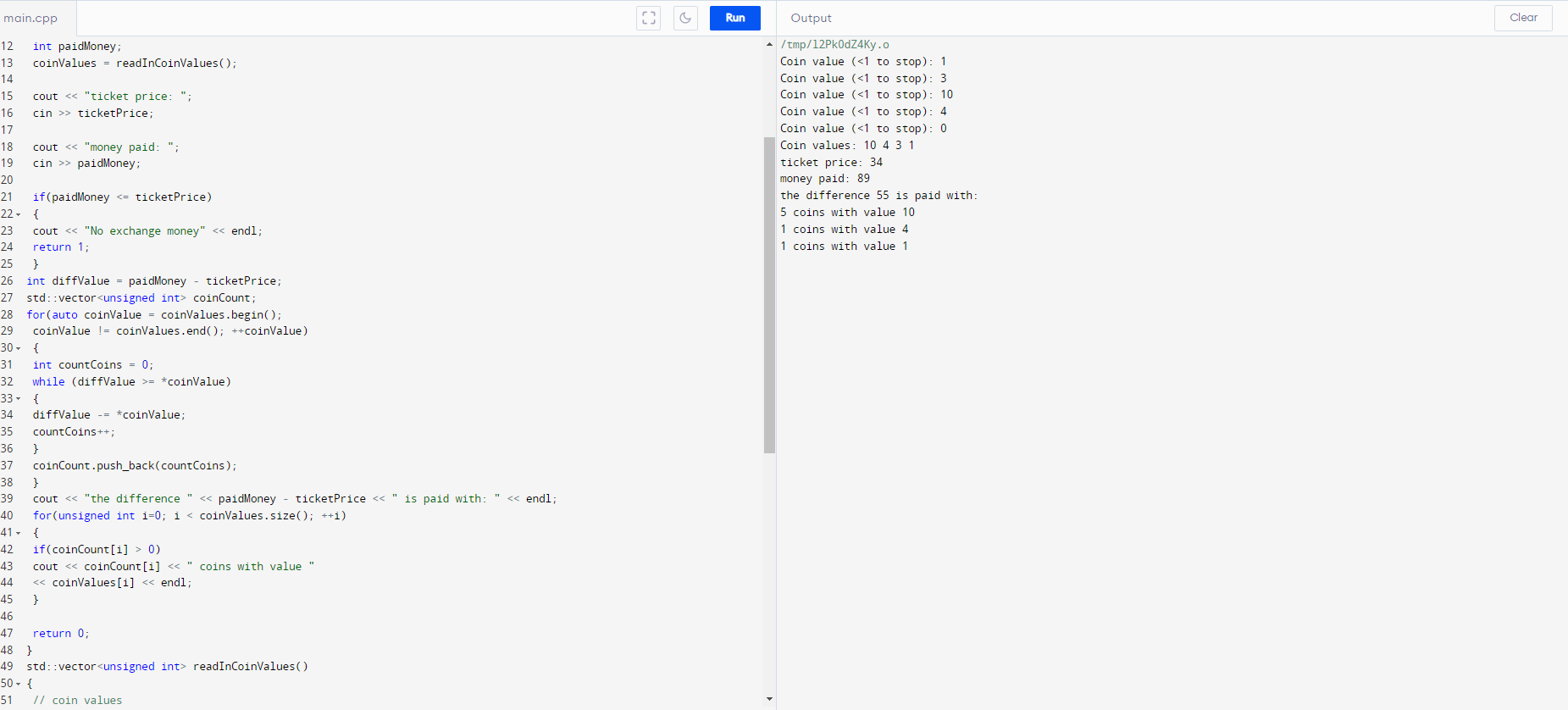
cout << "Coin values: ";

for(auto i : coinValues)

cout << i << " ";

cout << endl;

return coinValues;

} 

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

template <typename T>

class Add

{

T x;

public:

Add(T xx):x(xx){}

void operator()(T& e) const { e += x; }

};

template <typename T>

class PrintElements

{

public:

void operator()(T& elm) const { cout << elm << ' ';}

};

int main()

{

int n = 5; // cate valori sa aiba

vector<int> v; // cream un vector

for(int i = 0; i < n; i++) v.push\_back(i); //parcurgem vectorul

PrintElements<int> print\_it;

for\_each(v.begin(), v.end(), print\_it); cout << endl;//afiseaza toate elementele pana la <vector

for\_each(v.begin(), v.end(), Add<int>(5)); //oferim un numar de la care vrem sa plecam iar

for\_each(v.begin(), v.end(), print\_it); cout << endl; // afiseaza toate elementele de la numar pana la marimea vectorului

for\_each(v.begin(), v.end(), Add<int>(\*v.begin())); //dubleaza valoarea data

for\_each(v.begin(), v.end(), print\_it);//afiseaza elementele de la valoarea dublata pana la marimea vectorului

return 0;

}

