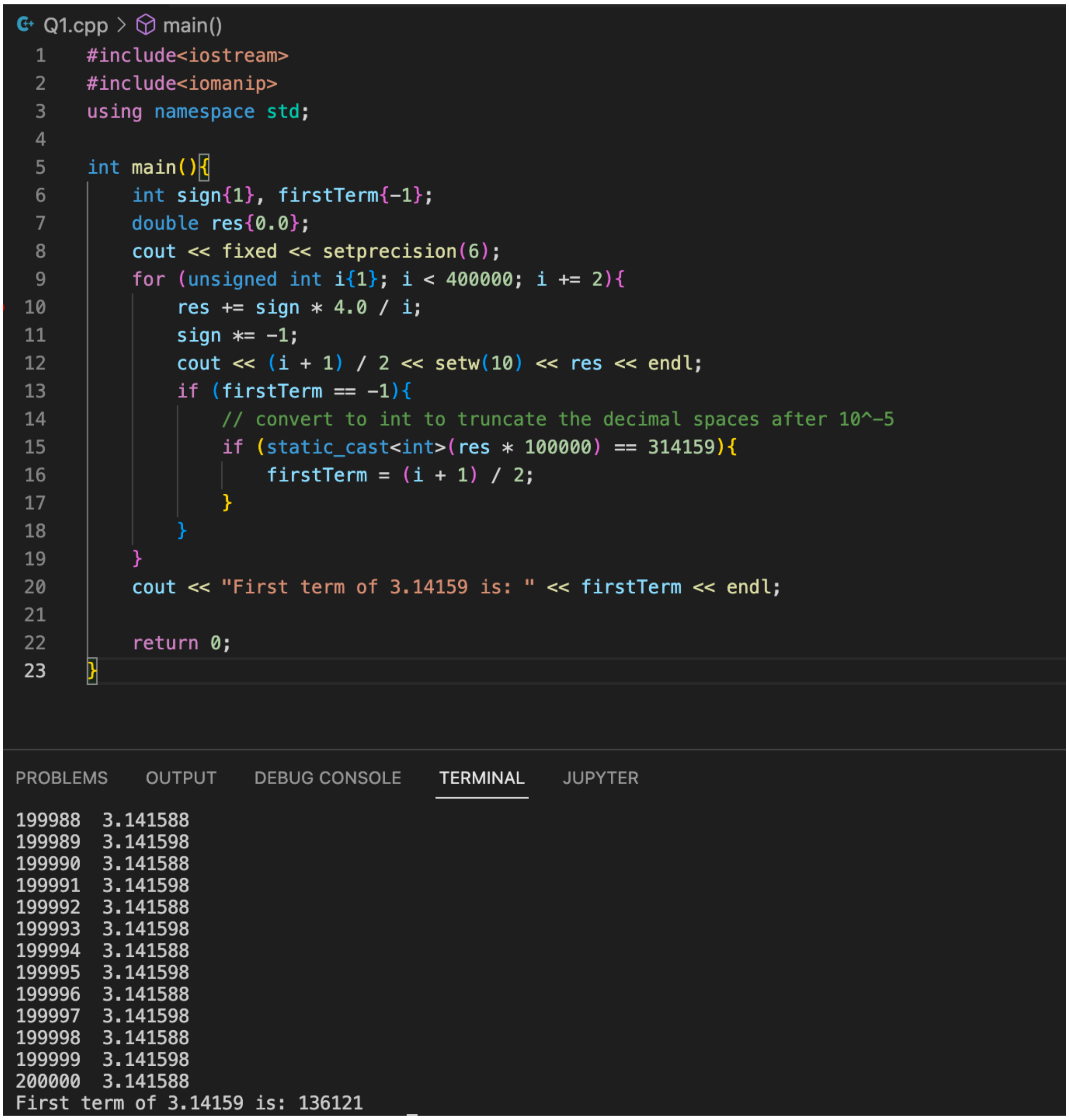
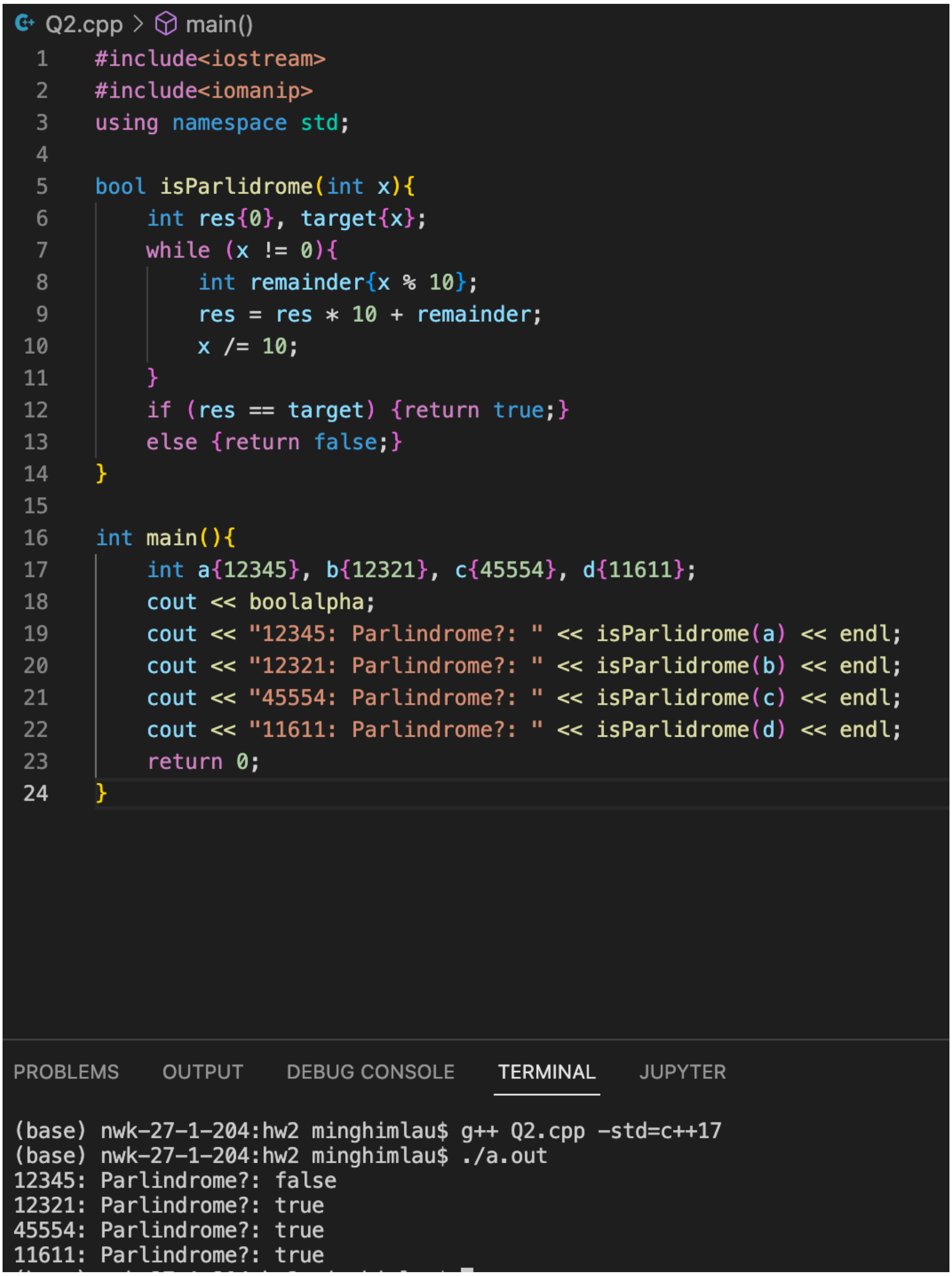
Q1.



```cpp
#include<iostream>
#include<iomanip>
using namespace std;

int main(){
    int sign{1}, firstTerm{-1};
    double res{0.0};
    cout << fixed << setprecision(6);
    for (unsigned int i{1}; i < 400000; i += 2){
        res += sign * 4.0 / i;
        sign *= -1;
        cout << (i + 1) / 2 << setw(10) << res << endl;
        if (firstTerm == -1){
            // convert to int to truncate the decimal spaces after 10^-5
            if (static_cast<int>(res * 100000) == 314159){
                firstTerm = (i + 1) / 2;
            }
        }
    }
    cout << "First term of 3.14159 is: " << firstTerm << endl;

    return 0;
}
```

```
199988  3.141588
199989  3.141598
199990  3.141588
199991  3.141598
199992  3.141588
199993  3.141598
199994  3.141588
199995  3.141598
199996  3.141588
199997  3.141598
199998  3.141588
199999  3.141598
200000  3.141588
First term of 3.14159 is: 136121
```

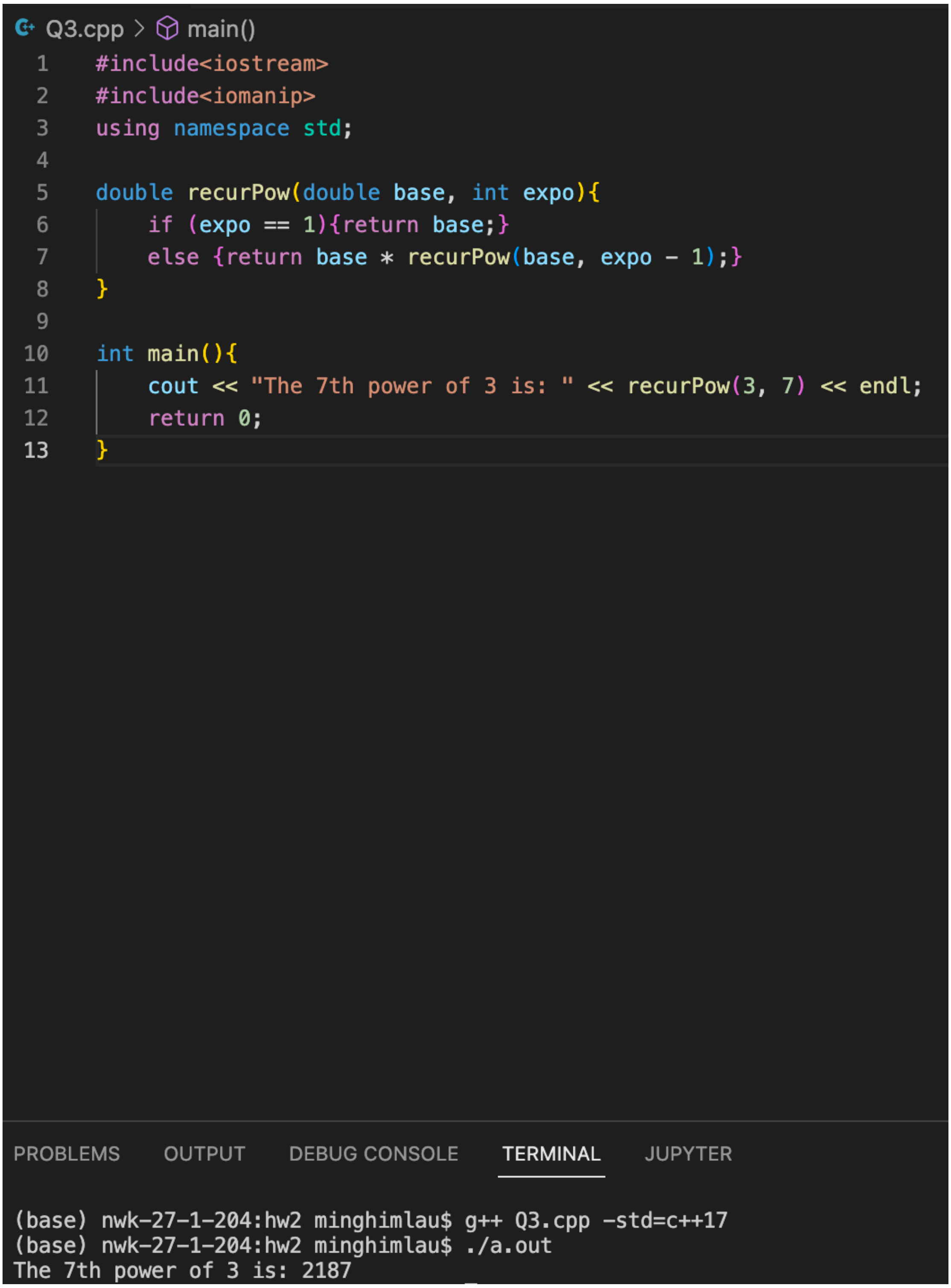
Q2.



```cpp
#include<iostream>
#include<iomanip>
using namespace std;

bool isParlidrome(int x){
    int res{0}, target{x};
    while (x != 0){
        int remainder{x % 10};
        res = res * 10 + remainder;
        x /= 10;
    }
    if (res == target) {return true;}
    else {return false;}
}

int main(){
    int a{12345}, b{12321}, c{45554}, d{11611};
    cout << boolalpha;
    cout << "12345: Parlindrome?: " << isParlidrome(a) << endl;
    cout << "12321: Parlindrome?: " << isParlidrome(b) << endl;
    cout << "45554: Parlindrome?: " << isParlidrome(c) << endl;
    cout << "11611: Parlindrome?: " << isParlidrome(d) << endl;
    return 0;
}
```

```
(base) nwk-27-1-204:hw2 minghimlau$ g++ Q2.cpp -std=c++17
(base) nwk-27-1-204:hw2 minghimlau$ ./a.out
12345: Parlindrome?: false
12321: Parlindrome?: true
45554: Parlindrome?: true
11611: Parlindrome?: true
```

Q3.

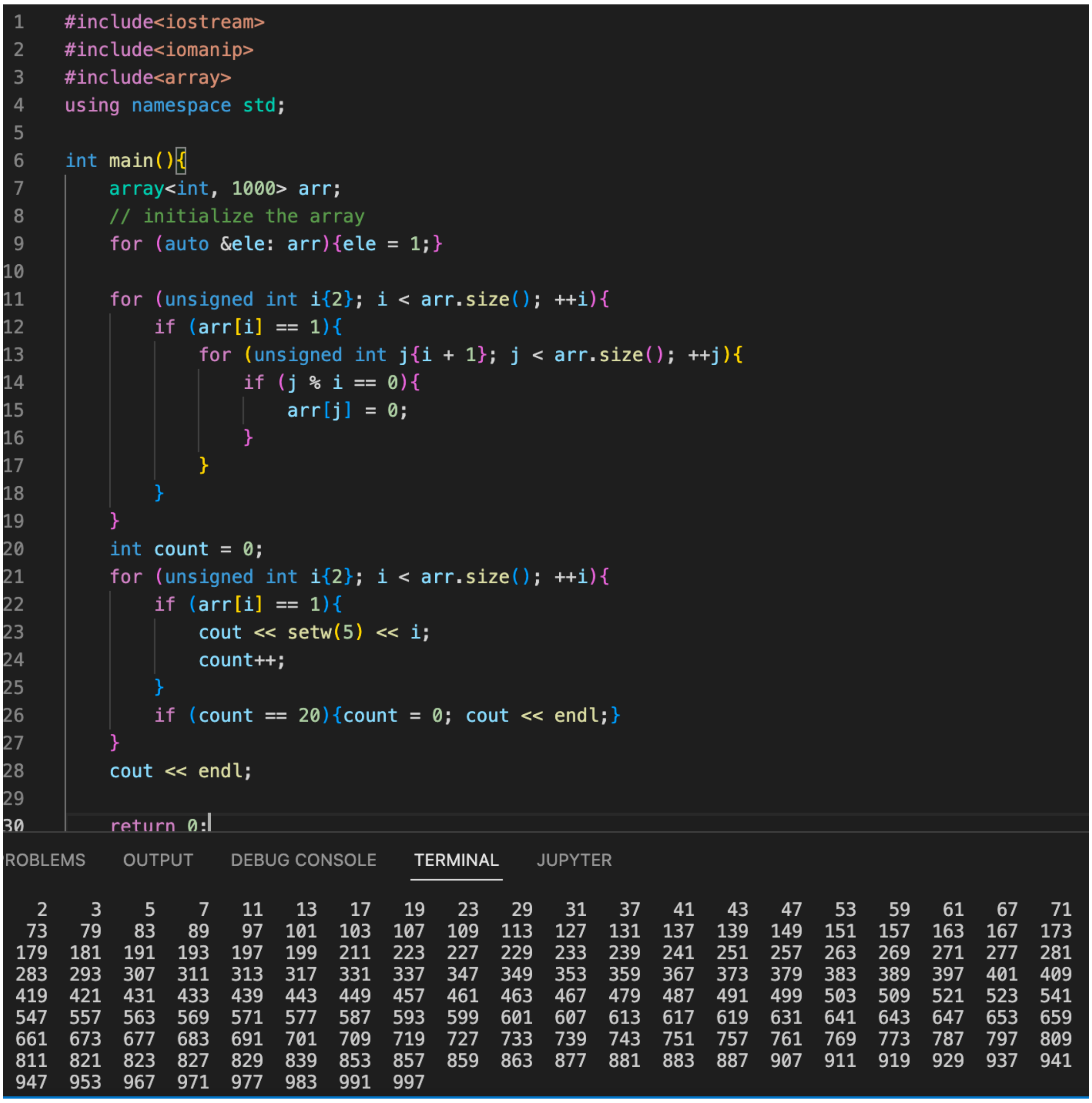


```cpp
#include<iostream>
#include<iomanip>
using namespace std;

double recurPow(double base, int expo){
    if (expo == 1){return base;}
    else {return base * recurPow(base, expo - 1);}
}

int main(){
    cout << "The 7th power of 3 is: " << recurPow(3, 7) << endl;
    return 0;
}
```

```
(base) nwk-27-1-204:hw2 minghimlau$ g++ Q3.cpp -std=c++17
(base) nwk-27-1-204:hw2 minghimlau$ ./a.out
The 7th power of 3 is: 2187
```

Q4. The function is a recursive function that takes an integer std::array as input and return the sum of all elements of the array. Thus, in this program, the function will return an integer 55 and the output will be: “The result is 55”.

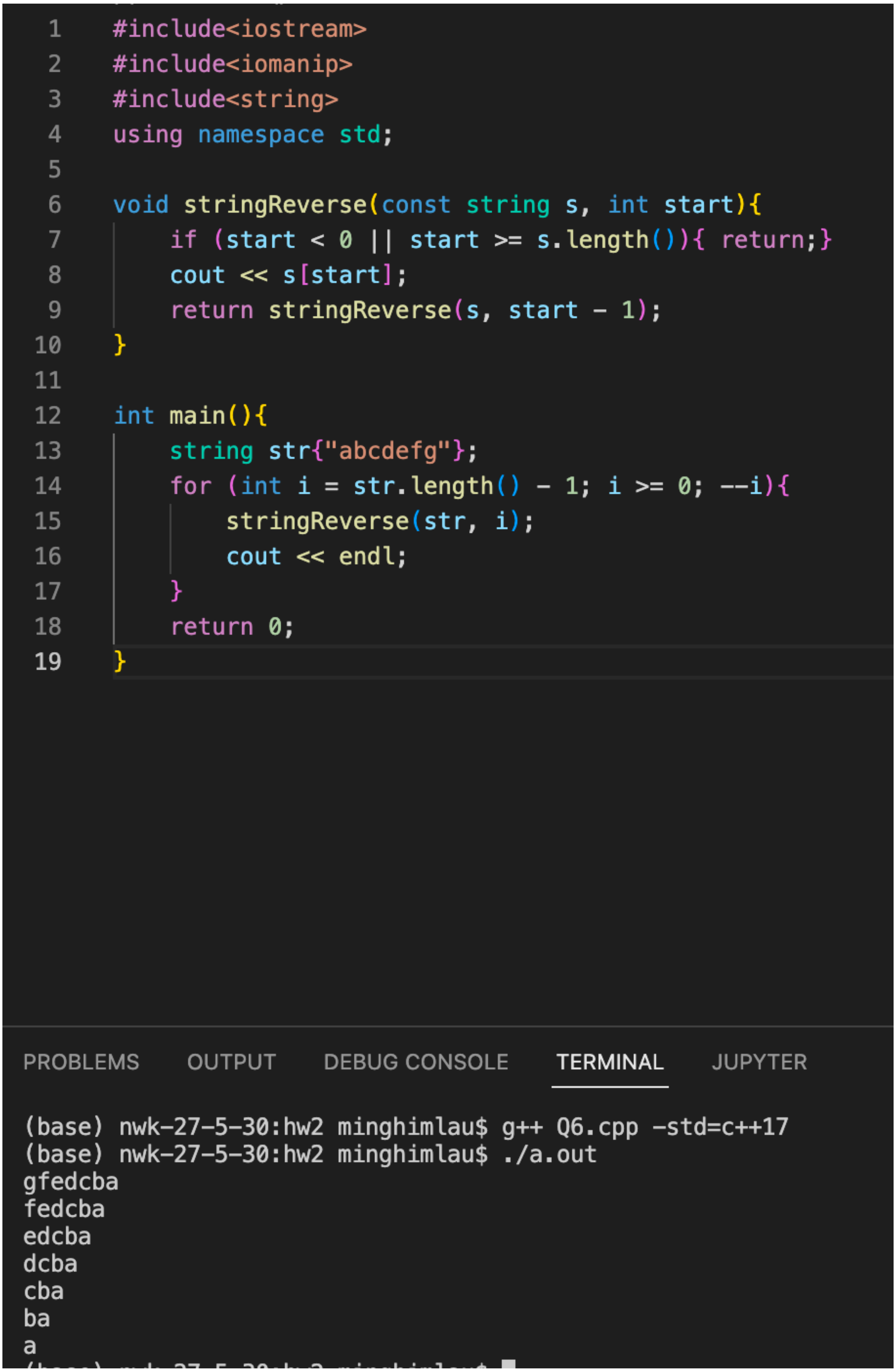
Q5.



```cpp
#include<iostream>
#include<iomanip>
#include<array>
using namespace std;

int main(){
    array<int, 1000> arr;
    // initialize the array
    for (auto &ele: arr){ele = 1;}

    for (unsigned int i{2}; i < arr.size(); ++i){
        if (arr[i] == 1){
            for (unsigned int j{i + 1}; j < arr.size(); ++j){
                if (j % i == 0){
                    arr[j] = 0;
                }
            }
        }
    }
    int count = 0;
    for (unsigned int i{2}; i < arr.size(); ++i){
        if (arr[i] == 1){
            cout << setw(5) << i;
            count++;
        }
        if (count == 20){count = 0; cout << endl;}
    }
    cout << endl;

    return 0;
```

```
    2    3    5    7   11   13   17   19   23   29   31   37   41   43   47   53   59   61   67   71
   73   79   83   89   97  101  103  107  109  113  127  131  137  139  149  151  157  163  167  173
  179  181  191  193  197  199  211  223  227  229  233  239  241  251  257  263  269  271  277  281
  283  293  307  311  313  317  331  337  347  349  353  359  367  373  379  383  389  397  401  409
  419  421  431  433  439  443  449  457  461  463  467  479  487  491  499  503  509  521  523  541
  547  557  563  569  571  577  587  593  599  601  607  613  617  619  631  641  643  647  653  659
  661  673  677  683  691  701  709  719  727  733  739  743  751  757  761  769  773  787  797  809
  811  821  823  827  829  839  853  857  859  863  877  881  883  887  907  911  919  929  937  941
  947  953  967  971  977  983  991  997
```

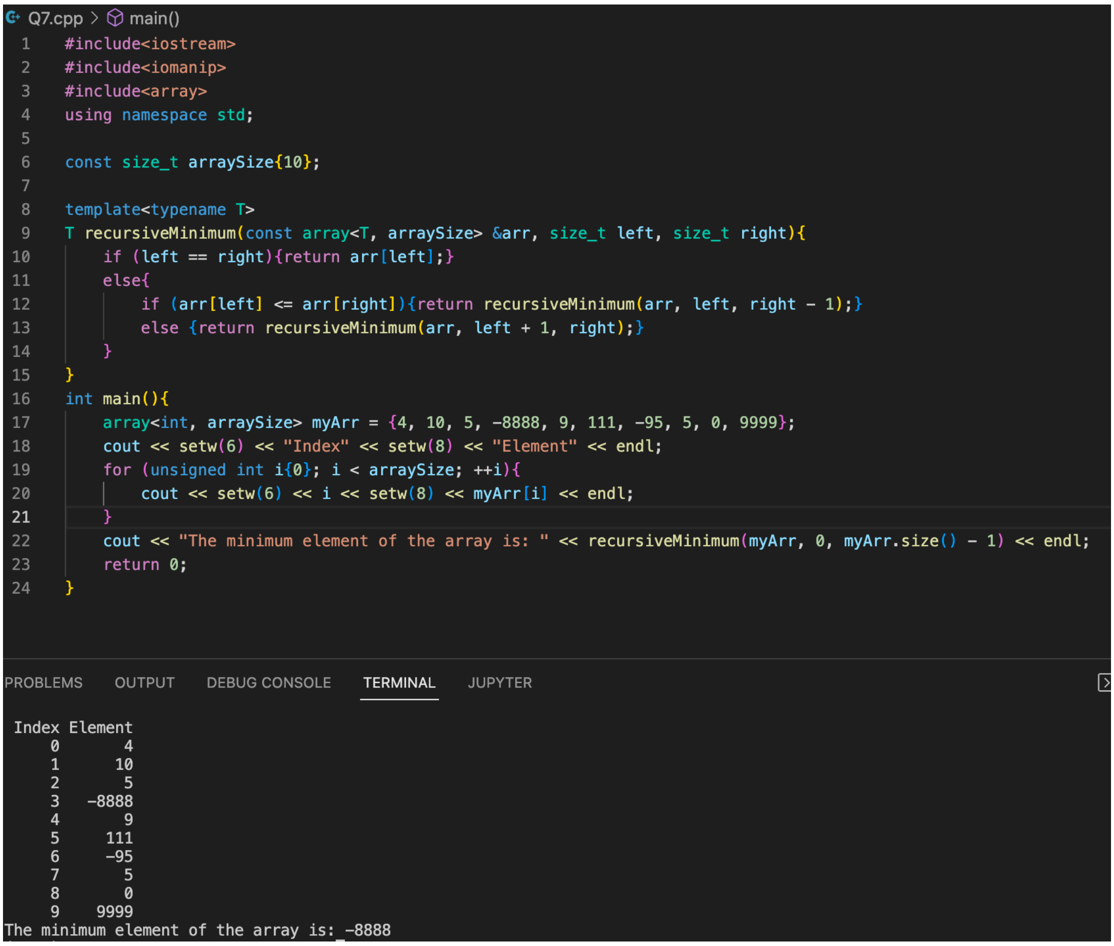
Q6.



```cpp
#include<iostream>
#include<iomanip>
#include<string>
using namespace std;

void stringReverse(const string s, int start){
    if (start < 0 || start >= s.length()){ return;}
    cout << s[start];
    return stringReverse(s, start - 1);
}

int main(){
    string str{"abcdefg"};
    for (int i = str.length() - 1; i >= 0; --i){
        stringReverse(str, i);
        cout << endl;
    }
    return 0;
}
```

```
(base) nwk-27-5-30:hw2 minghimlau$ g++ Q6.cpp -std=c++17
(base) nwk-27-5-30:hw2 minghimlau$ ./a.out
gfedcba
fedcba
edcba
dcba
cba
ba
a
(base) nwk-27-5-30:hw2 minghimlau$ 
```

Q7.



```cpp
#include<iostream>
#include<iomanip>
#include<array>
using namespace std;

const size_t arraySize{10};

template<typename T>
T recursiveMinimum(const array<T, arraySize> &arr, size_t left, size_t right){
    if (left == right){return arr[left];}
    else{
        if (arr[left] <= arr[right]){return recursiveMinimum(arr, left, right - 1);}
        else {return recursiveMinimum(arr, left + 1, right);}
    }
}
int main(){
    array<int, arraySize> myArr = {4, 10, 5, -8888, 9, 111, -95, 5, 0, 9999};
    cout << setw(6) << "Index" << setw(8) << "Element" << endl;
    for (unsigned int i{0}; i < arraySize; ++i){
        cout << setw(6) << i << setw(8) << myArr[i] << endl;
    }
    cout << "The minimum element of the array is: " << recursiveMinimum(myArr, 0, myArr.size() - 1) << endl;
    return 0;
}
```

```
 Index Element
     0       4
     1      10
     2       5
     3   -8888
     4       9
     5     111
     6     -95
     7       5
     8       0
     9    9999
The minimum element of the array is: -8888
```