NAME	**:**	**SHIFANA.J**
ROLL.NO	**:**	**18BIT047**
DEPARTMENT	**:**	**INFORMATION TECHNOLOGY**
INSTITUTION	**:**	**SRI KRISHNA ARTS AND SCIECE COLLEGE**
E- MAIL	**:**	**Shifanayasmine0@gmail.com**

BURROWS – WHEELER TRANSFORM

Problem Statement :

Perform Burrows-Wheeler transform or block-sorting compression on the below-mentioned sentence.

“Curneu MedTech Innovation is a health care technology firm based at Heidelberg, Germany. We work on a motive of building affordable and innovative healthcare solutions that address the clinical needs thereby bringing better lives for the needy.”

Definition:

The **Burrows–Wheeler transform** (**BWT**, also called **block-sorting compression**) rearranges character string runs of similar characters. This is useful for compression, since it tends to be easy to compress a string that has runs of repeated characters by techniques such as front transform and run – length encoding. More importantly, the transformation is *reversible*, without needing to store any additional data except the position of the first original character. The BWT is thus a "free" method of improving the efficiency of text compression algorithms, costing only some extra computation.

FINAL CODE:

```
#include <algorithm>
#include <iostream>
#include <vector>

const int STX = 0x02;
const int ETX = 0x03;

void rotate(std::string &n) {
    char t = n[n.length() - 1];
    for (int i = n.length() - 1; i > 0; i--) {
        n[i] = n[i - 1];
    }
    n[0] = t;
}

std::string bwt(const std::string &s) {
    for (char c : s) {
        if (c == STX || c == ETX) {
            throw std::runtime_error("Input can't contain STX or ETX");
        }
    }

    std::string h;
    h += STX;
    h += s;
    h += ETX;

    std::vector<std::string> table;
    for (size_t i = 0; i <h.length(); i++) {
        table.push_back(h);
        rotate(h);
    }
    std::sort(table.begin(), table.end());

    std::string out;
    for (auto &s : table) {
        out += s[s.length() - 1];
    }
    return out;
}

std::string ibwt(const std::string &f) {
    int len = f.length();
    std::vector<std::string> table(len);
    for (int i = 0; i < len; i++) {
        for (int j = 0; j < len; j++) {
            table[j] = f[j] + table[j];
        }
        std::sort(table.begin(), table.end());
    }
    for (auto &row : table) {
        if (row[row.length() - 1] == ETX) {
            return row.substr(1, row.length() - 2);
        }
    }
    return {};
}

std::string makePrintable(const std::string &s) {
    auto ls = s;
    for (auto &c : ls) {
        if (c == STX) {
            c = '^';
        }
    }
    return ls;
}

int main() {
    auto tests = {
        "Curneu MedTech Innovation is a health care technology firm based at Heidelberg, Germany.",
        "We work on a motive of building affordable and innovative healthcare solutions that address the clinical needs thereby bringing better lives for the needy"
    };

    for (auto &test : tests) {
        std::cout << makePrintable(test) << "\n";
        std::cout << " Burrows - Wheeler Transform -> ";

        std::string t;
        try {
            t = bwt(test);
            std::cout << makePrintable(t) << "\n";
        } catch (std::runtime_error &e) {
            std::cout << "Error " << e.what() << "\n";
        }
    }


    return 0;
}
```

EXPLANATION OF CODE:

- #include <algorithm> : is a collection of functions especially designed to be used on ranges of elements.
- #include< upstream> : is used to get standard input and output.
- #include<vector> : is used to resize itself automatically when an element is inserted or deleted.
- STX and ETX : start of text character and End of text character is used to run code easily and less error – prone .

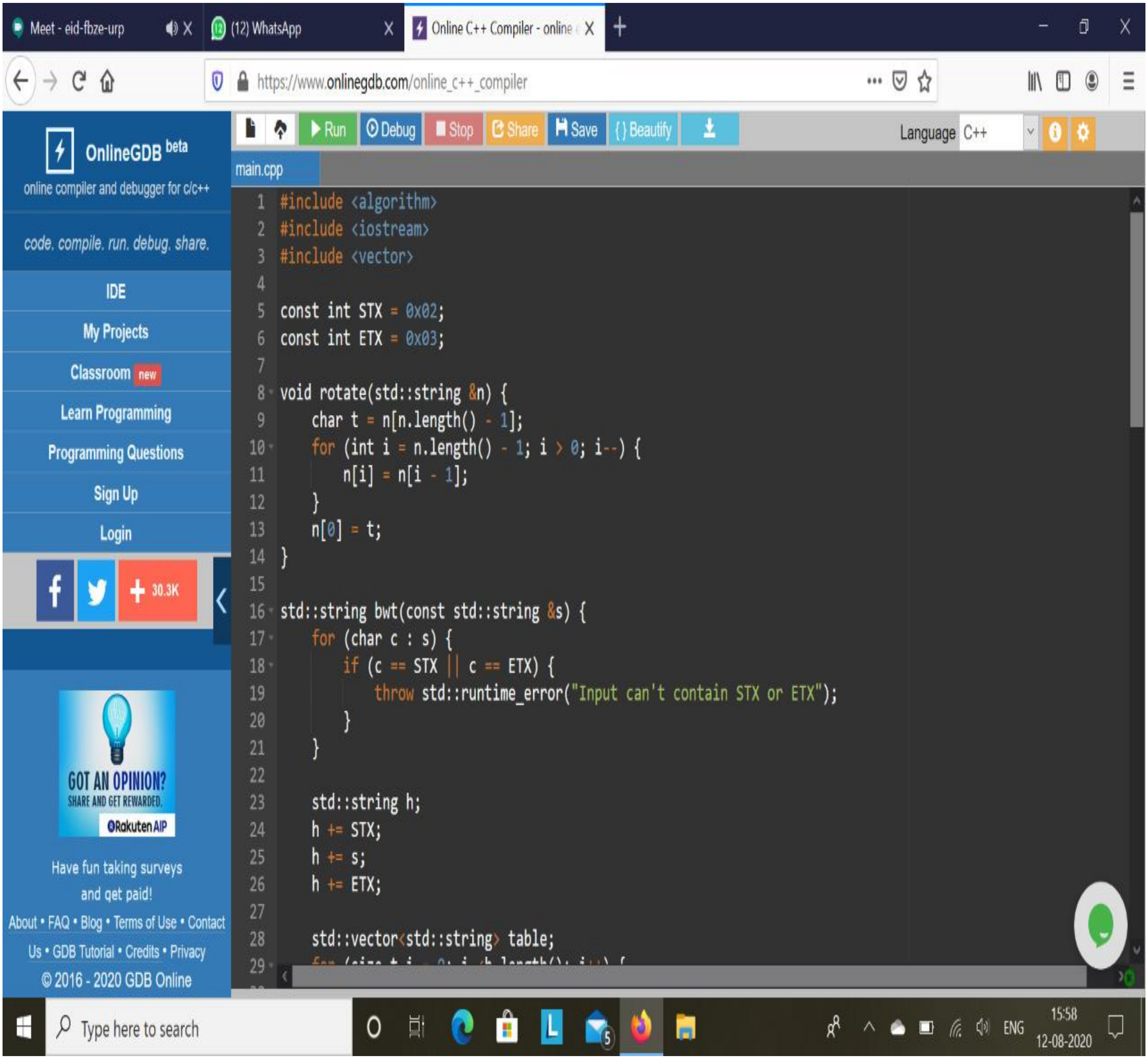
Three major steps involved in the code:

1. Cyclic rotation
2. Sorting
3. Lexicographical order

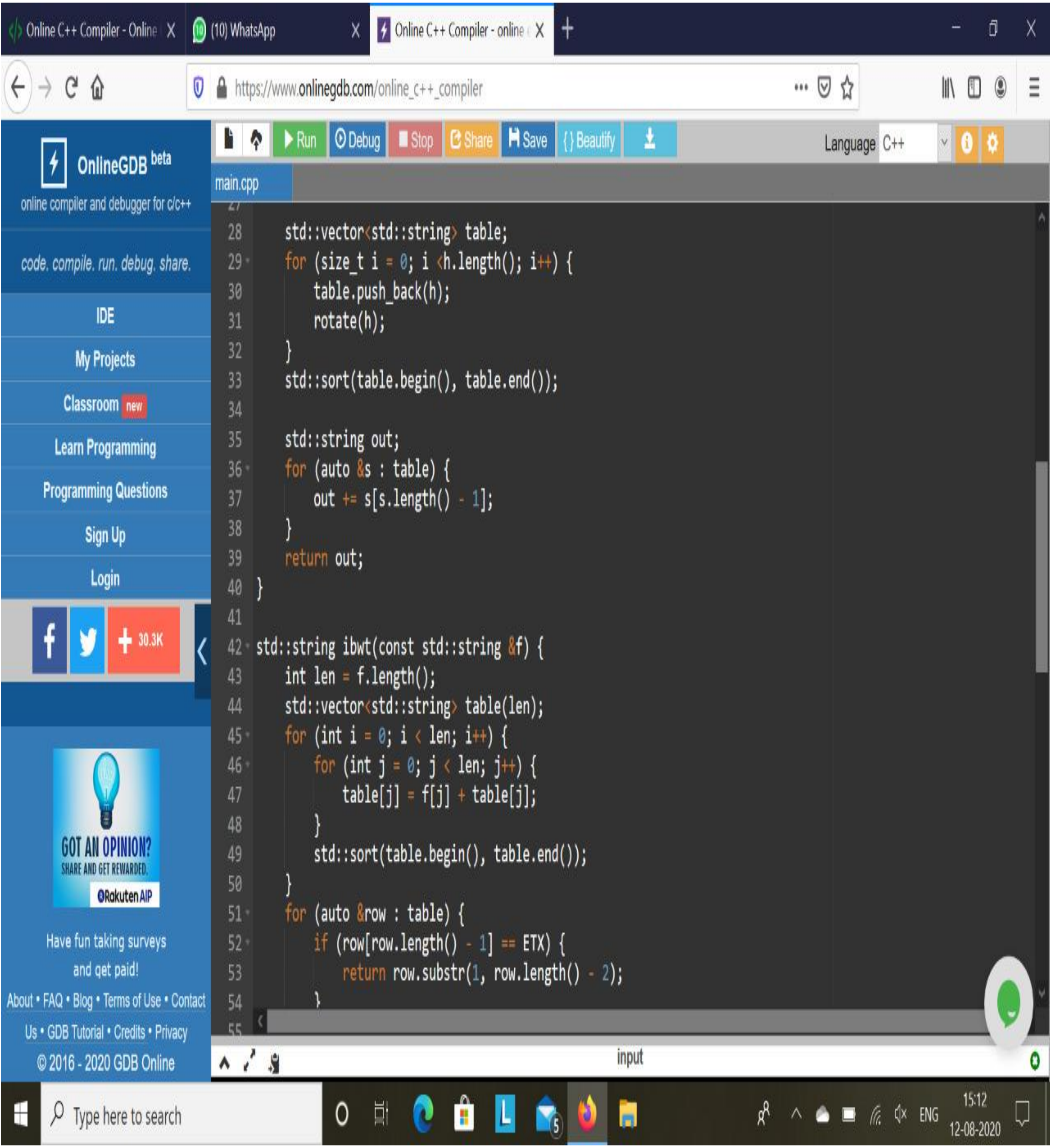
- To perform these Operations for loop and if statements are used .
- VOID ROTATE : is used to rotate the order of elements.
- STD:: STRING : it is a way to represent sequence of characters as an object of class.
- SCOPE RESOLUTION OPERATOR : is used for accessing the character.
- PUSH : is used for moving the characters.
- MakePrintable : statement is used for printing the statements and output statement.

- &(and) : is used for accessing the address or memory space of the character.
- Inside main() function the input statements are given.
- Trycatch: is used if the given input statements are in characters it will follow burrows-wheeler transform algorithm orelse it will throw exception.

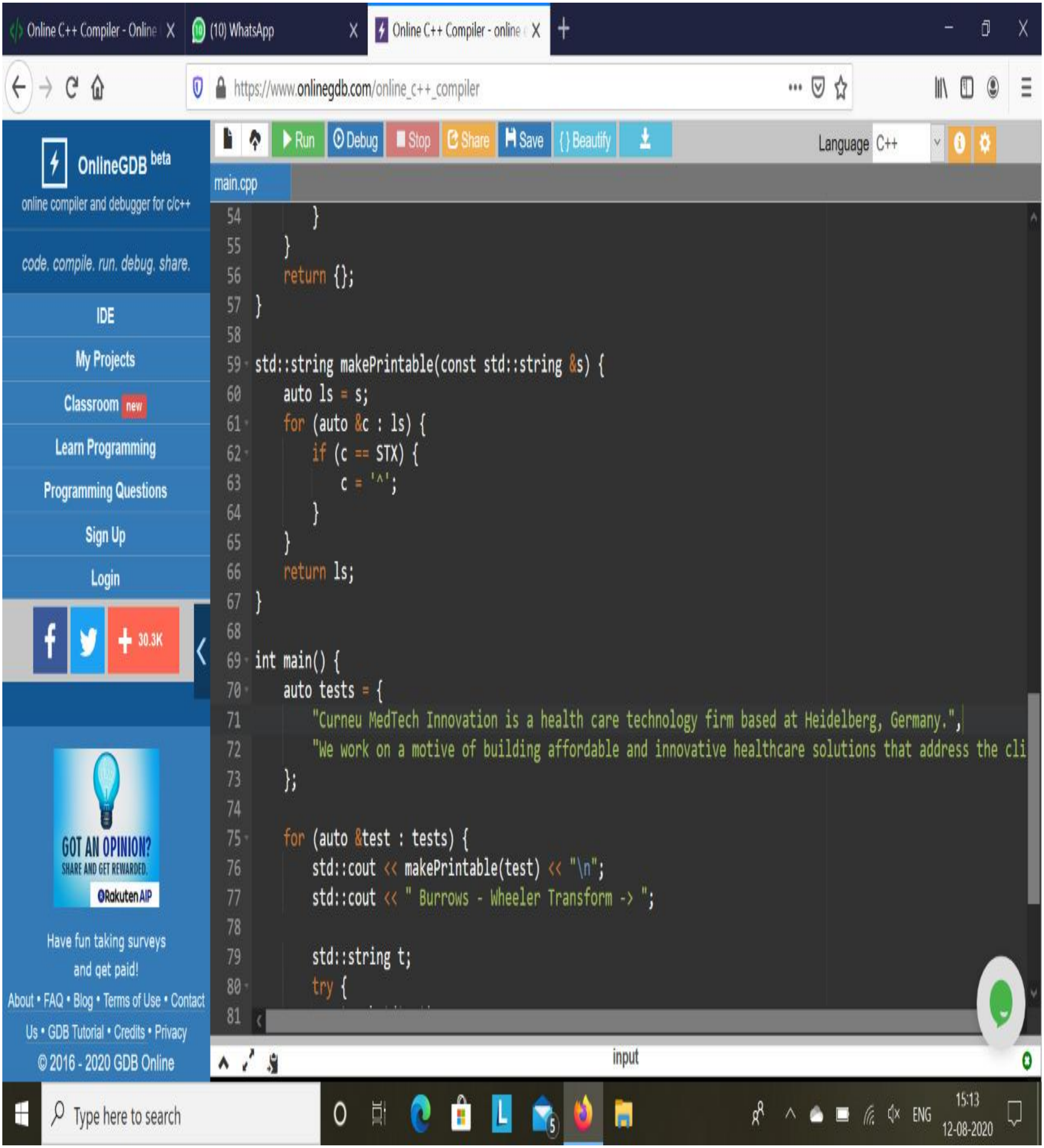
PROGRAM:



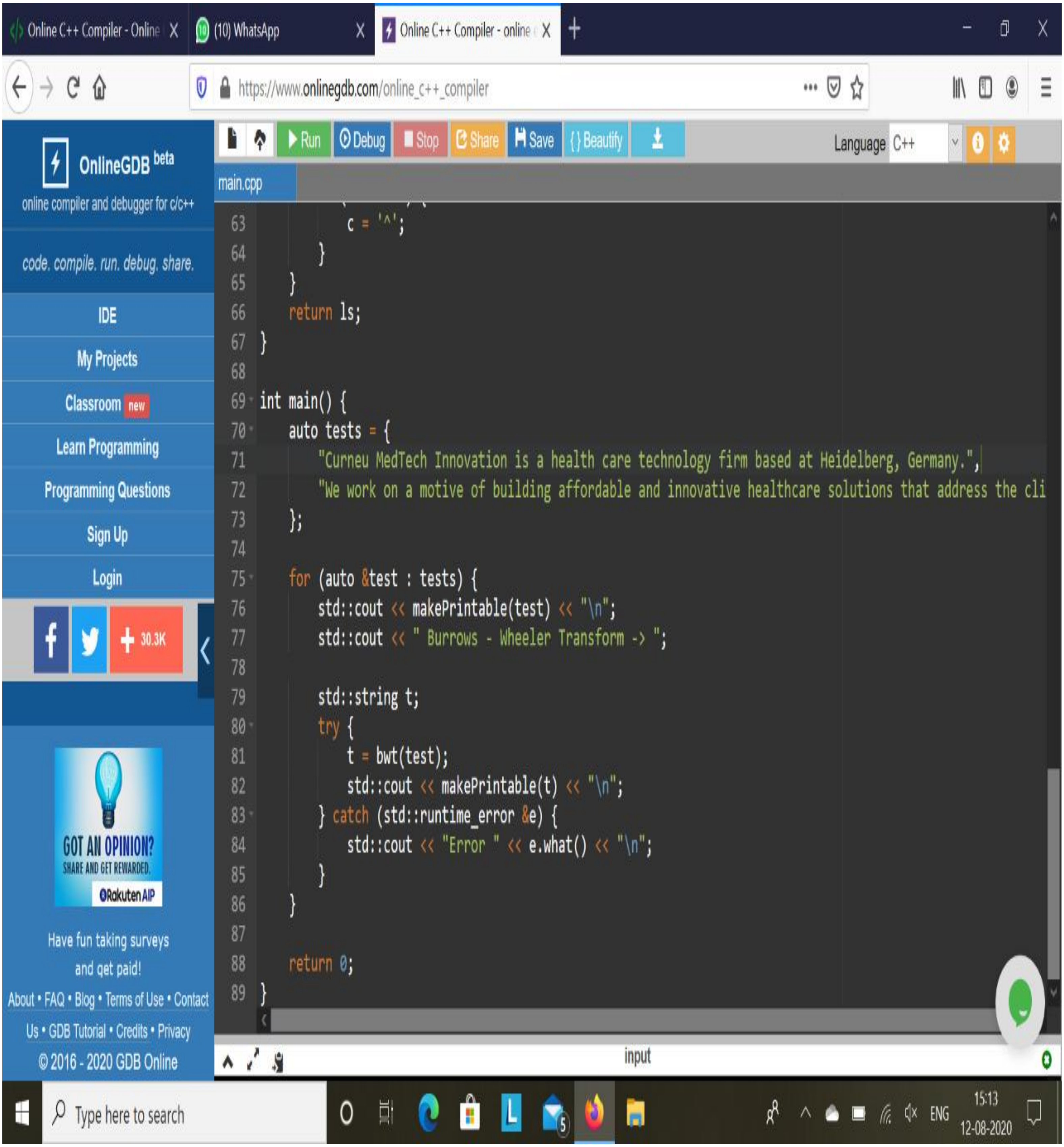
```cpp
#include <algorithm>
#include <iostream>
#include <vector>

const int STX = 0x02;
const int ETX = 0x03;

void rotate(std::string &n) {
    char t = n[n.length() - 1];
    for (int i = n.length() - 1; i > 0; i--) {
        n[i] = n[i - 1];
    }
    n[0] = t;
}

std::string bwt(const std::string &s) {
    for (char c : s) {
        if (c == STX || c == ETX) {
            throw std::runtime_error("Input can't contain STX or ETX");
        }
    }

    std::string h;
    h += STX;
    h += s;
    h += ETX;

    std::vector<std::string> table;
    for (size_t i = 0; i < h.length(); i++) {
```


Online C++ Compiler - Online
(10) WhatsApp
Online C++ Compiler - online
https://www.onlinegdb.com/online_c++_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
30.3K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2020 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C++
main.cpp
input
Type here to search
ENG
15:12
12-08-2020

```cpp
    std::vector<std::string> table;
    for (size_t i = 0; i <h.length(); i++) {
        table.push_back(h);
        rotate(h);
    }
    std::sort(table.begin(), table.end());

    std::string out;
    for (auto &s : table) {
        out += s[s.length() - 1];
    }
    return out;
}

std::string ibwt(const std::string &f) {
    int len = f.length();
    std::vector<std::string> table(len);
    for (int i = 0; i < len; i++) {
        for (int j = 0; j < len; j++) {
            table[j] = f[j] + table[j];
        }
        std::sort(table.begin(), table.end());
    }
    for (auto &row : table) {
        if (row[row.length() - 1] == ETX) {
            return row.substr(1, row.length() - 2);
        }
```

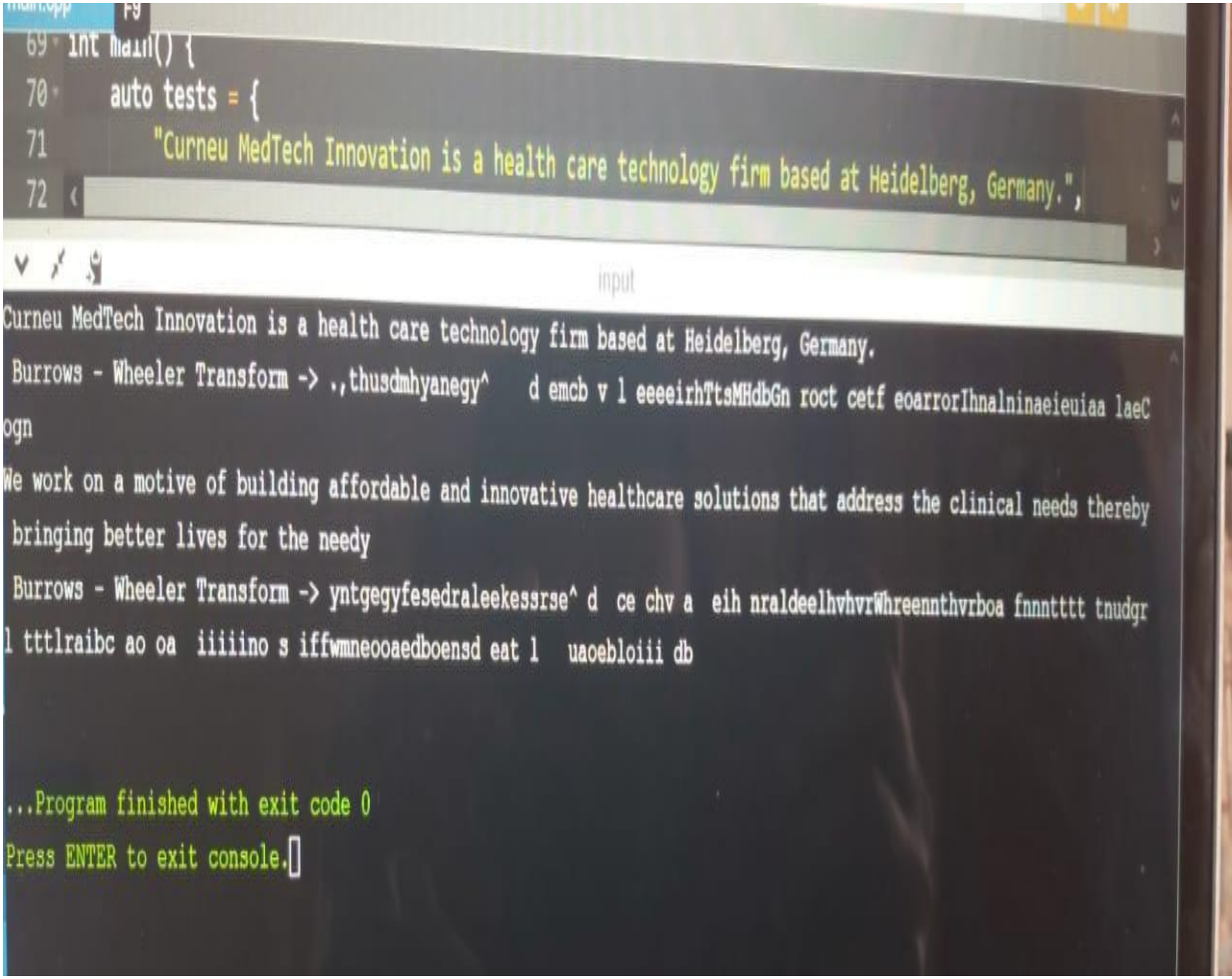

Online C++ Compiler - Online
(10) WhatsApp
Online C++ Compiler - online
https://www.onlinegdb.com/online_c++_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
30.3K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2020 GDB Online
Run
Debug
Stop
Share
Save
{} Beautify
Language C++
main.cpp
input
Type here to search
ENG
15:13
12-08-2020

```cpp
        }
    }
    return {};
}

std::string makePrintable(const std::string &s) {
    auto ls = s;
    for (auto &c : ls) {
        if (c == STX) {
            c = '^';
        }
    }
    return ls;
}

int main() {
    auto tests = {
        "Curneu MedTech Innovation is a health care technology firm based at Heidelberg, Germany.",
        "We work on a motive of building affordable and innovative healthcare solutions that address the cli
    };

    for (auto &test : tests) {
        std::cout << makePrintable(test) << "\n";
        std::cout << " Burrows - Wheeler Transform -> ";

        std::string t;
        try {
```


Online C++ Compiler - Online
(10) WhatsApp
Online C++ Compiler - online
https://www.onlinegdb.com/online_c++_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
30.3K
GOT AN OPINION?
SHARE AND GET REWARDED.
Rakuten AIP
Have fun taking surveys and get paid!
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2020 GDB Online
Run
Debug
Stop
Share
Save
{} Beautify
Language C++
main.cpp
input
Type here to search
ENG
15:13
12-08-2020

```
            c = '^';
        }
    }
    return ls;
}

int main() {
    auto tests = {
        "Curneu MedTech Innovation is a health care technology firm based at Heidelberg, Germany.",
        "We work on a motive of building affordable and innovative healthcare solutions that address the cli
    };

    for (auto &test : tests) {
        std::cout << makePrintable(test) << "\n";
        std::cout << " Burrows - Wheeler Transform -> ";

        std::string t;
        try {
            t = bwt(test);
            std::cout << makePrintable(t) << "\n";
        } catch (std::runtime_error &e) {
            std::cout << "Error " << e.what() << "\n";
        }
    }

    return 0;
}
```

OUTPUT:



```
int main() {
    auto tests = {
        "Curneu MedTech Innovation is a health care technology firm based at Heidelberg, Germany.",
```

```
Curneu MedTech Innovation is a health care technology firm based at Heidelberg, Germany.
 Burrows - Wheeler Transform -> .,thusdmhyanegy^    d emcb v l eeeeirhTtsMHdbGn roct cetf eoarrorIhnalninaeieuiaa laeC
ogn
We work on a motive of building affordable and innovative healthcare solutions that address the clinical needs thereby
 bringing better lives for the needy
 Burrows - Wheeler Transform -> yntgegyfesedraleekessrse^ d  ce chv a  eih nraldeelhvhvrWhreennthvrboa fnnntttt tnudgr
l tttlraibc ao oa  iiiiino s iffwmneooaedboensd eat l   uaoebloiii db



...Program finished with exit code 0
Press ENTER to exit console.
```