

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

The screenshot shows a web browser window with the URL [https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler). The browser's address bar and tabs are visible at the top. The OnlineGDB interface includes a sidebar on the left with navigation links: IDE, My Projects, Classroom (marked 'new'), Learn Programming, Programming Questions, Sign Up, and Login. Below these are social media icons for Facebook and Twitter, and a '30.3K' badge. A Rakuten AIP advertisement is also present in the sidebar.

The main area displays a C++ program in a dark-themed editor. The code is as follows:

```
1 #include <algorithm>
2 #include <iostream>
3 #include <vector>
4
5 const int STX = 0x02;
6 const int ETX = 0x03;
7
8 void rotate(std::string &n) {
9     char t = n[n.length() - 1];
10    for (int i = n.length() - 1; i > 0; i--) {
11        n[i] = n[i - 1];
12    }
13    n[0] = t;
14 }
15
16 std::string bwt(const std::string &s) {
17     for (char c : s) {
18         if (c == STX || c == ETX) {
19             throw std::runtime_error("Input can't contain STX or ETX");
20         }
21     }
22
23     std::string h;
24     h += STX;
25     h += s;
26     h += ETX;
27
28     std::vector<std::string> table;
29     for (size_t i = 0; i < h.length(); i++) {
```

The Windows taskbar is visible at the bottom of the screen, showing the Start button, a search bar, and several application icons. The system clock in the bottom right corner indicates the time is 15:58 on 12-08-2020.



```
27  
28 std::vector<std::string> table;  
29 for (size_t i = 0; i < h.length(); i++) {  
30     table.push_back(h);  
31     rotate(h);  
32 }  
33 std::sort(table.begin(), table.end());  
34  
35 std::string out;  
36 for (auto &s : table) {  
37     out += s[s.length() - 1];  
38 }  
39 return out;  
40 }
```

```
41  
42 std::string ibwt(const std::string &f) {  
43     int len = f.length();  
44     std::vector<std::string> table(len);  
45     for (int i = 0; i < len; i++) {  
46         for (int j = 0; j < len; j++) {  
47             table[j] = f[j] + table[j];  
48         }  
49         std::sort(table.begin(), table.end());  
50     }  
51     for (auto &row : table) {  
52         if (row[row.length() - 1] == ETX) {  
53             return row.substr(1, row.length() - 2);  
54         }  
55     }
```

input

**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

f + 30.3K

```
main.cpp
54 }
55 }
56 return {};
57 }
58
59 std::string makePrintable(const std::string &s) {
60     auto ls = s;
61     for (auto &c : ls) {
62         if (c == STX) {
63             c = '^';
64         }
65     }
66     return ls;
67 }
68
69 int main() {
70     auto tests = {
71         "Curneu MedTech Innovation is a health care technology firm based at Heidelberg, Germany.",
72         "We work on a motive of building affordable and innovative healthcare solutions that address the cli
73     };
74
75     for (auto &test : tests) {
76         std::cout << makePrintable(test) << "\n";
77         std::cout << " Burrows - Wheeler Transform -> ";
78
79         std::string t;
80         try {
81             <
```

**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

f t + 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

```
main.cpp
63     c = '^';
64 }
65 }
66 return ls;
67 }
68
69 int main() {
70     auto tests = {
71         "Curneu MedTech Innovation is a health care technology firm based at Heidelberg, Germany.",
72         "We work on a motive of building affordable and innovative healthcare solutions that address the cli
73     };
74
75     for (auto &test : tests) {
76         std::cout << makePrintable(test) << "\n";
77         std::cout << " Burrows - Wheeler Transform -> ";
78
79         std::string t;
80         try {
81             t = bwt(test);
82             std::cout << makePrintable(t) << "\n";
83         } catch (std::runtime_error &e) {
84             std::cout << "Error " << e.what() << "\n";
85         }
86     }
87
88     return 0;
89 }
```



## OUTPUT:

```
main.cpp 19
69 int main() {
70     auto tests = {
71         "Curneu MedTech Innovation is a health care technology firm based at Heidelberg, Germany.",
72         <
input
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 -> yntgegyfessedraleekessrse^ d ce chv a eih nraldeelhvhrWhreennthvrboa fnnntttt tnudgr
l ttltlraibc ao oa iiiino s iffwmneooaedboensd eat l uaoebloiii db

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