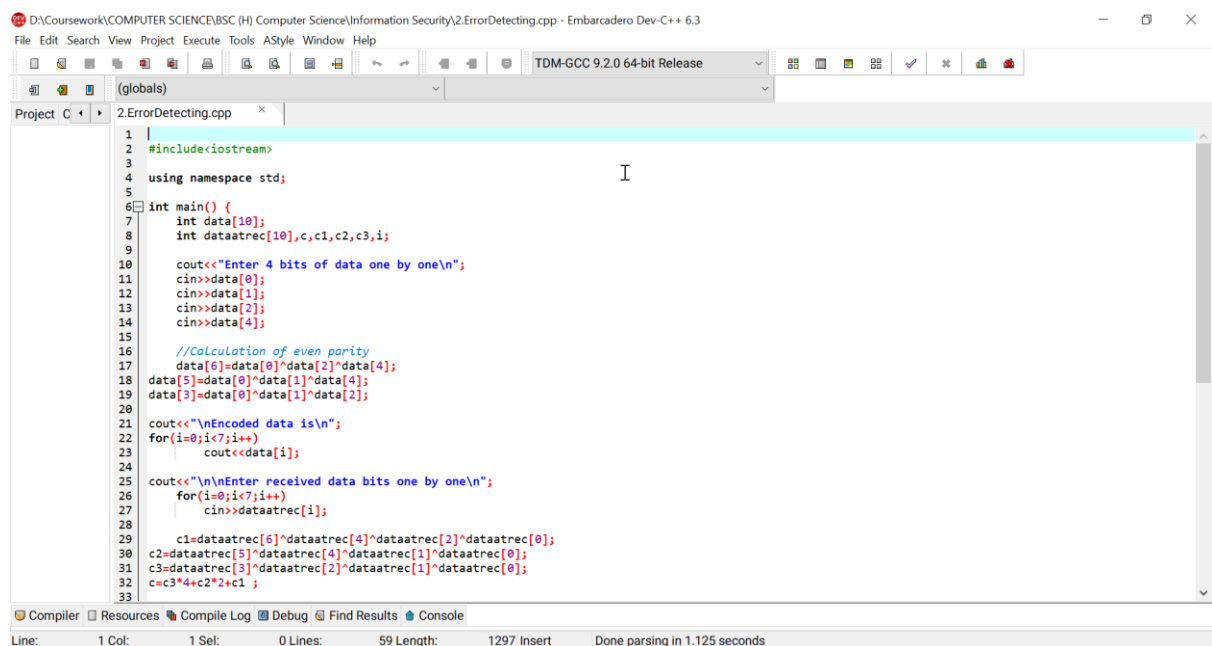


INFORMATION SECURITY

PRACTICAL FILE

Shivam Saxena
BSc(H) Computer Science
20201423

1. Implement the error detecting code



The screenshot shows the Embarcadero Dev-C++ IDE with a C++ program for error detection. The program is titled "2.ErrorDetecting.cpp" and is located at "D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\2.ErrorDetecting.cpp". The code is as follows:

```
1 |
2 | #include<iostream>
3 |
4 | using namespace std;
5 |
6 | int main() {
7 |     int data[10];
8 |     int dataatrec[10],c1,c2,c3,i;
9 |
10 |     cout<<"Enter 4 bits of data one by one\n";
11 |     cin>>data[0];
12 |     cin>>data[1];
13 |     cin>>data[2];
14 |     cin>>data[4];
15 |
16 |     //Calculation of even parity
17 |     data[6]=data[0]^data[2]^data[4];
18 |     data[5]=data[0]^data[1]^data[4];
19 |     data[3]=data[0]^data[1]^data[2];
20 |
21 |     cout<<"\nEncoded data is\n";
22 |     for(i=0;i<7;i++)
23 |         cout<<data[i];
24 |
25 |     cout<<"\n\nEnter received data bits one by one\n";
26 |     for(i=0;i<7;i++)
27 |         cin>>dataatrec[i];
28 |
29 |     c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
30 |     c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
31 |     c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
32 |     c=c3*4+c2*2+c1 ;
33 | }
```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), a toolbar, and a status bar at the bottom showing "Line: 1 Col: 1 Sel: 0 Lines: 59 Length: 1297 Insert Done parsing in 1.125 seconds".

```
D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\2.ErrorDetecting.cpp - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project C++ 2.ErrorDetecting.cpp
28
29 c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
30 c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
31 c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
32 c=c3*4+c2*2+c1;
33
34 if(c==0) {
35 cout<<"\nNo error while transmission of data\n";
36 }
37 else {
38 cout<<"\nError on position "<<c;
39 cout<<"\nData sent : ";
40 for(i=0;i<7;i++)
41 cout<<data[i];
42
43 cout<<"\nData received : ";
44 for(i=0;i<7;i++)
45 cout<<dataatrec[i];
46
47 cout<<"\nCorrect message is\n";
48
49 //if erroneous bit is 0 we complement it else vice versa
50 if(dataatrec[7-c]==0)
51 dataatrec[7-c]=1;
52 else
53 dataatrec[7-c]=0;
54 for (i=0;i<7;i++) {
55 cout<<dataatrec[i];
56 }
57 }
58 return 0;
59 }
Compiler Resources Compile Log Debug Find Results Console
Line: 1 Col: 1 Sel: 0 Lines: 59 Length: 1297 Insert Done parsing in 1.125 seconds
```

OUTPUT

```
D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\2.ErrorDetecting.exe
Enter 4 bits of data one by one
1
0
1
0
Encoded data is
1010010
Enter received data bits one by one
1
1
1
0
1
1
1
Error on position 4
Data sent : 1010010
Data received : 1110111
Correct message is
1111111
-----
Process exited after 27.22 seconds with return value 0
Press any key to continue . . .
```

3. Implement caesar cipher substitution operation.

The image shows two screenshots of a C++ IDE (Embarcadero Dev-C++ 6.3) implementing a Caesar cipher substitution operation. The first screenshot shows the `encrypt` and `decrypt` functions. The second screenshot shows the `main` function and the `decrypt` function.

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 char str[100],str1[100];
4 void encrypt(int key)
5 {
6     for(int i=0;i<strlen(str);i++)
7     {
8         if((int)str[i]+key>122)
9         {
10             str[i] = (char)((int)str[i]+key-26);
11             continue;
12         }
13         else
14             (str[i]=(char)((int)str[i]+key));
15     }
16     cout<<"Encrypted String is : "<<endl;puts(str);
17 }
18 void decrypt(int key)
19 {
20     for(int i=0;i<strlen(str);i++)
21     {
22         if((int)str[i]-key<97) str1[i] = (char)((int)str[i]-key+26);
23         else str1[i]=(char)((int)str[i]-key);
24     }
25     cout<<"Key is : "<<key <<"\nDecrypted String is : ";puts(str1);
26 }
27 int main()
```

Compiler Output:

```
- Output Filename: D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\3.CaesarCipher.exe
- Output Size: 2.99188899993896 MiB
- Compilation Time: 3.00s
```

Line: 1 Col: 1 Sel: 0 Lines: 44 Length: 1017 Insert Done parsing in 0.016 seconds

```
19 {
20     for(int i=0;i<strlen(str);i++)
21     {
22         if((int)str[i]-key<97) str1[i] = (char)((int)str[i]-key+26);
23         else str1[i]=(char)((int)str[i]-key);
24     }
25     cout<<"Key is : "<<key <<"\nDecrypted String is : ";puts(str1);
26 }
27 int main()
28 {
29     int key;
30     cout<<"Enter String : "<<endl;
31     gets(str);
32     cout<<"Enter Key : "<<endl;
33     cin >> key;
34     encrypt(key);
35     decrypt(key);
36     cout<<"Attack starts : "<<endl;
37     for(int i=0;i<26;i++)
38     {
39         decrypt(i);
40     }
41     cout<<endl<<"Made by: AADITYA KEDIYAL "<<endl;
42     return 0;
43 }
44 }
```

Compiler Output:

```
- Output Filename: D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\3.CaesarCipher.exe
- Output Size: 2.99188899993896 MiB
- Compilation Time: 3.00s
```

Line: 1 Col: 1 Sel: 0 Lines: 44 Length: 1017 Insert Done parsing in 0.016 seconds

OUTPUT

D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\3.CaesarCipher.exe

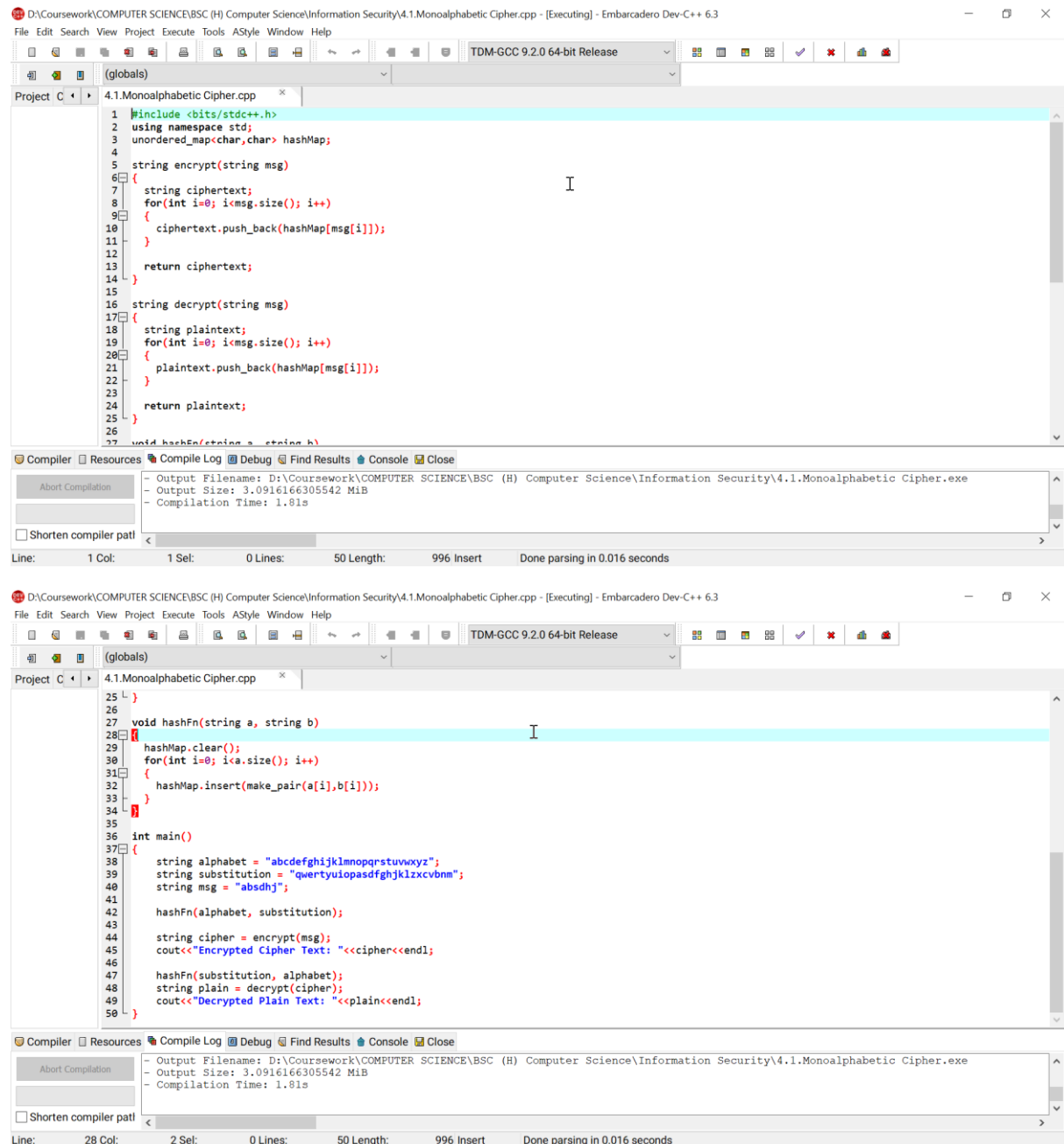
```
Enter String :  
iamaadi  
Enter Key :  
4  
Encrypted String is :  
meqeehm  
Key is : 4  
Decrypted String is : iamaadi  
Attack starts :  
Key is : 0  
Decrypted String is : meqeehm  
Key is : 1  
Decrypted String is : ldpddgl  
Key is : 2  
Decrypted String is : kcoccfk  
Key is : 3  
Decrypted String is : jbnbbbj  
Key is : 4  
Decrypted String is : iamaadi  
Key is : 5  
Decrypted String is : hzlzzch  
Key is : 6  
Decrypted String is : gykyybg  
Key is : 7  
Decrypted String is : fxjxxaf  
Key is : 8  
Decrypted String is : ewiwwze  
Key is : 9  
Decrypted String is : dvhvvyd  
Key is : 10  
Decrypted String is : cuguuxc  
Key is : 11  
Decrypted String is : btfttwb  
Key is : 12  
Decrypted String is : asessva  
Key is : 13  
Decrypted String is : zrdrruz  
Key is : 14  
Decrypted String is : yqcqqty  
Key is : 15  
Decrypted String is : xpbppsx
```

D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\3.CaesarCipher.exe

```
Decrypted String is : fxjxxaf  
Key is : 8  
Decrypted String is : ewiwwze  
Key is : 9  
Decrypted String is : dvhvvyd  
Key is : 10  
Decrypted String is : cuguuxc  
Key is : 11  
Decrypted String is : btfttwb  
Key is : 12  
Decrypted String is : asessva  
Key is : 13  
Decrypted String is : zrdrruz  
Key is : 14  
Decrypted String is : yqcqqty  
Key is : 15  
Decrypted String is : xpbppsx  
Key is : 16  
Decrypted String is : woaoorw  
Key is : 17  
Decrypted String is : vnznqv  
Key is : 18  
Decrypted String is : umymmpu  
Key is : 19  
Decrypted String is : tixllot  
Key is : 20  
Decrypted String is : skwkkns  
Key is : 21  
Decrypted String is : rjvjvjr  
Key is : 22  
Decrypted String is : qiuiilq  
Key is : 23  
Decrypted String is : phthhkp  
Key is : 24  
Decrypted String is : ogsggjo  
Key is : 25  
Decrypted String is : nfrffin  
  
Made by: AADITYA KEDIYAL  
-----
```

4.

4.1 . Implement monoalphabetic cipher substitution operation.



```
1 #include <bits/stdc++.h>
2 using namespace std;
3 unordered_map<char, char> hashMap;
4
5 string encrypt(string msg)
6 {
7     string ciphertext;
8     for(int i=0; i<msg.size(); i++)
9     {
10         ciphertext.push_back(hashMap[msg[i]]);
11     }
12     return ciphertext;
13 }
14
15 string decrypt(string msg)
16 {
17     string plaintext;
18     for(int i=0; i<msg.size(); i++)
19     {
20         plaintext.push_back(hashMap[msg[i]]);
21     }
22     return plaintext;
23 }
24
25 void hashFn(string a, string b)
26 {
27     hashMap.clear();
28     for(int i=0; i<a.size(); i++)
29     {
30         hashMap.insert(make_pair(a[i], b[i]));
31     }
32 }
33
34 int main()
35 {
36     string alphabet = "abcdefghijklmnopqrstuvwxyz";
37     string substitution = "qwertyuiopasdfghjklzxcvbnm";
38     string msg = "absdhj";
39
40     hashFn(alphabet, substitution);
41
42     string cipher = encrypt(msg);
43     cout<<"Encrypted Cipher Text: "<<cipher<<endl;
44
45     hashFn(substitution, alphabet);
46     string plain = decrypt(cipher);
47     cout<<"Decrypted Plain Text: "<<plain<<endl;
48 }
49
50
```

Compiler | Resources | Compile Log | Debug | Find Results | Console | Close

Output Filename: D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\4.1.Monoalphabetic Cipher.exe
Output Size: 3.0916166305542 Mib
Compilation Time: 1.81s

Line: 28 Col: 2 Sel: 0 Lines: 50 Length: 996 Insert Done parsing in 0.016 seconds

OUTPUT

```
D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\4.1.Monoalphabetic Cipher.exe
Encrypted Cipher Text: qwlrip
Decrypted Plain Text: absdhj
-----
Process exited after 0.09004 seconds with return value 0
Press any key to continue . . .
```

4.2 Implement polyalphabetic cipher substitution operation

```
D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\4.2.Polyalphabetic Cipher.cpp - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project C++ 4.2.Polyalphabetic Cipher.cpp
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class Vig {
5 public:
6     string k;
7     Vig(string k) {
8         for (int i = 0; i < k.size(); ++i) {
9             if (k[i] >= 'A' && k[i] <= 'Z')
10                 this->k += k[i];
11             else if (k[i] >= 'a' && k[i] <= 'z')
12                 this->k += k[i] + 'A' - 'a';
13         }
14     }
15     string encryption(string t) {
16         string output;
17         for (int i = 0, j = 0; i < t.length(); ++i) {
18             char c = t[i];
19             if (c >= 'a' && c <= 'z')
20                 c += 'A' - 'a';
21             else if (c < 'A' || c > 'Z')
22                 continue;
23             output += (c + k[j] - 2 * 'A') % 26 + 'A'; //added 'A' to bring it in range of ASCII alphabet [ 65-90 | A-Z ]
24             j = (j + 1) % k.length();
25         }
26         return output;
27     }
28 };
29 int main() {
30     string k, t;
31     cout << "Enter key: ";
32     getline(cin, k);
33     cout << "Enter text: ";
34     getline(cin, t);
35     Vig v(k);
36     string enc = v.encryption(t);
37     cout << "Encrypted text: " << enc << endl;
38     return 0;
39 }
```

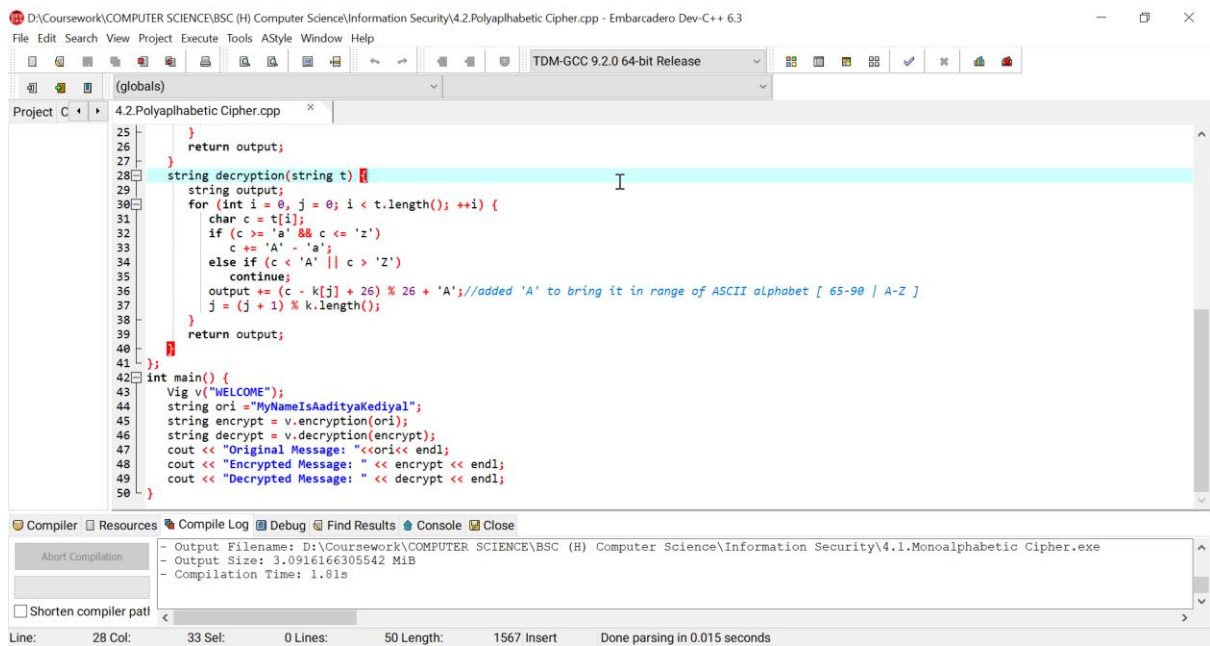
Compiler Resources Compile Log Debug Find Results Console Close

Abort Compilation

Output Filename: D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\4.1.Monoalphabetic Cipher.exe
Output Size: 3.0916166305542 MiB
Compilation Time: 1.81s

Shorten compiler path

Line: 1 Col: 1 Sel: 0 Lines: 50 Length: 1567 Insert Done parsing in 0.015 seconds



```
25 }
26 return output;
27 }
28 string decryption(string t)
29 {
30     string output;
31     for (int i = 0, j = 0; i < t.length(); ++i) {
32         char c = t[i];
33         if (c >= 'a' && c <= 'z')
34             c += 'A' - 'a';
35         else if (c < 'A' || c > 'Z')
36             continue;
37         output += (c - k[j] + 26) % 26 + 'A'; //added 'A' to bring it in range of ASCII alphabet [ 65-90 | A-Z ]
38         j = (j + 1) % k.length();
39     }
40     return output;
41 }
42 int main() {
43     Vig v("WELCOME");
44     string ori = "MyNameIsAadityaKediyal";
45     string encrypt = v.encryption(ori);
46     string decrypt = v.decryption(encrypt);
47     cout << "Original Message: " << ori << endl;
48     cout << "Encrypted Message: " << encrypt << endl;
49     cout << "Decrypted Message: " << decrypt << endl;
50 }
```

Compiler Resources Compile Log Debug Find Results Console Close

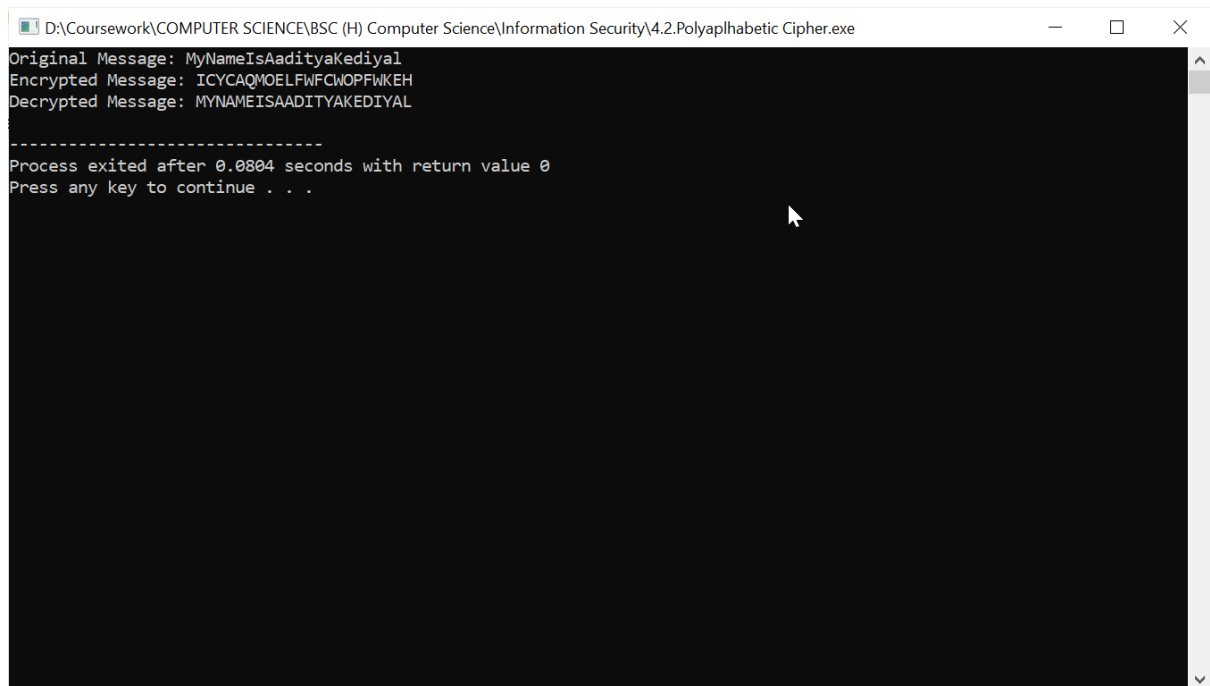
Abort Compilation

Output Filename: D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\4.1.Monoalphabetic Cipher.exe
Output Size: 3.0916166305542 MiB
Compilation Time: 1.81s

Shorten compiler path

Line: 28 Col: 33 Sel: 0 Lines: 50 Length: 1567 Insert Done parsing in 0.015 seconds

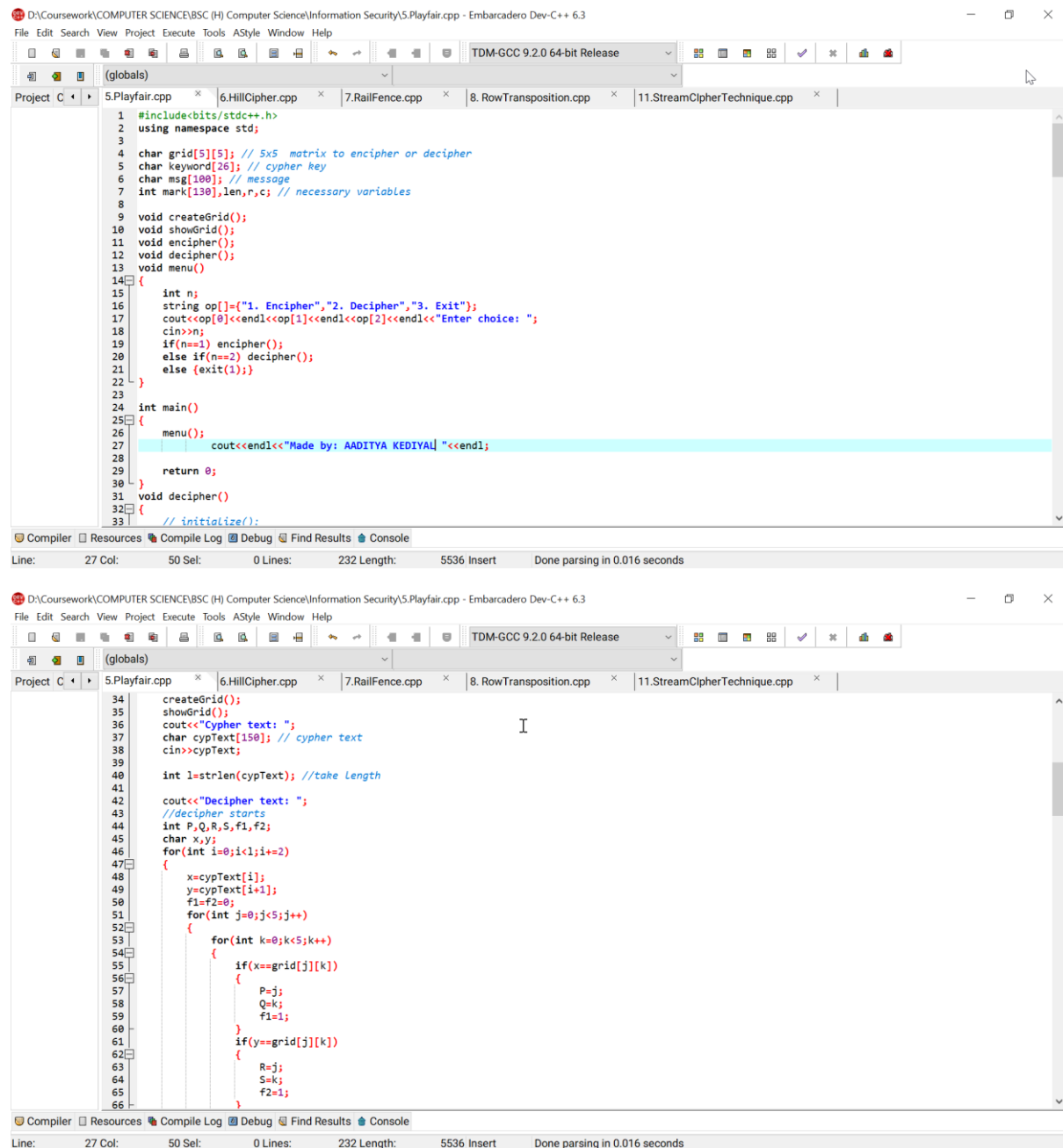
OUTPUT



```
D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\4.2.Polyaplhabetic Cipher.exe
Original Message: MyNameIsAadityaKediyal
Encrypted Message: ICYCAQMOELFWFCWOPFWKEH
Decrypted Message: MYNAMEISAADITYAKEDIYAL

-----
Process exited after 0.0804 seconds with return value 0
Press any key to continue . . .
```

5. Implement playfair cipher substitution operation.



The image shows two screenshots of a C++ IDE (Embarcadero Dev-C++ 6.3) implementing the Playfair cipher substitution operation. The first screenshot shows the main menu and initialization code, while the second screenshot shows the deciphering logic.

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 char grid[5][5]; // 5x5 matrix to encipher or decipher
5 char keyword[26]; // cypher key
6 char msg[100]; // message
7 int mark[130],len,r,c; // necessary variables
8
9 void createGrid();
10 void showGrid();
11 void encipher();
12 void decipher();
13 void menu();
14 {
15     int n;
16     string op[3]={"1. Encipher","2. Decipher","3. Exit"};
17     cout<<op[0]<<endl<<op[1]<<endl<<op[2]<<endl<<"Enter choice: ";
18     cin>>n;
19     if(n==1) encipher();
20     else if(n==2) decipher();
21     else {exit(1);}
22 }
23
24 int main()
25 {
26     menu();
27     cout<<endl<<"Made by: AADITYA KEDIYAL" <<endl;
28
29     return 0;
30 }
31 void decipher()
32 {
33     // initialize();
```

Compiler Resources Compile Log Debug Find Results Console
Line: 27 Col: 50 Sel: 0 Lines: 232 Length: 5536 Insert Done parsing in 0.016 seconds

```
34 createGrid();
35 showGrid();
36 cout<<"Cypher text: ";
37 char cypText[150]; // cypher text
38 cin>>cypText;
39
40 int l=strlen(cypText); //take Length
41
42 cout<<"Decipher text: ";
43 //decipher starts
44 int P,Q,R,S,f1,f2;
45 char x,y;
46 for(int i=0;i<l;i+=2)
47 {
48     x=cypText[i];
49     y=cypText[i+1];
50     f1=f2=0;
51     for(int j=0;j<5;j++)
52     {
53         for(int k=0;k<5;k++)
54         {
55             if(x==grid[j][k])
56             {
57                 P=j;
58                 Q=k;
59                 f1=1;
60             }
61             if(y==grid[j][k])
62             {
63                 R=j;
64                 S=k;
65                 f2=1;
66             }
```

Compiler Resources Compile Log Debug Find Results Console
Line: 27 Col: 50 Sel: 0 Lines: 232 Length: 5536 Insert Done parsing in 0.016 seconds

D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\5.Playfair.cpp - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C++

5.Playfair.cpp x 6.HillCipher.cpp x 7.RailFence.cpp x 8.RowTransposition.cpp x 11.StreamCipherTechnique.cpp x

```
66
67     } if(f1 && f2) break;
68
69     } if(f1 && f2) break;
70
71     if(P==R) //same row
72     {
73         if(Q==0) cout<<grid[P][4];
74         else cout<<grid[P][Q-1];
75         if(S==0) cout<<grid[R][4];
76         else cout<<grid[R][S-1];
77     }
78     else if(Q==S) // same column
79     {
80         if(P==0) cout<<grid[4][Q];
81         else cout<<grid[P-1][Q];
82         if(R==0) cout<<grid[4][S];
83         else cout<<grid[R-1][S];
84     }
85     else //opposite corner
86     {
87         cout<<grid[P][S]<<grid[R][Q];
88     }
89     }
90     cout<<endl<<endl;
91     menu();
92 }
93
94 void encipher()
95 {
96     // initialize();
97     createGrid();
98     showGrid();
99     cout<<"Message to cypher: ";
```

Compiler Resources Compile Log Debug Find Results Console

Line: 27 Col: 50 Sel: 0 Lines: 232 Length: 5536 Insert Done parsing in 0.016 seconds

D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\5.Playfair.cpp - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C++

5.Playfair.cpp x 6.HillCipher.cpp x 7.RailFence.cpp x 8.RowTransposition.cpp x 11.StreamCipherTechnique.cpp x

```
97 showGrid();
98 cout<<"Message to cypher: ";
99 gets(msg);
100 int l=strlen(msg); // msg length
101 char reqText[150]; //generate required text to encipher
102 int in=0,j=0;
103 for(int i=0;i<l;i++)
104 {
105     j=i+1;
106     if(msg[i]!=' ') //ignore all space from string
107     {
108         i++;
109         j++;
110     }
111     if(msg[j]!=' ') j++; //ignore space
112     if(toupper(msg[i])=='j') msg[i]='i'; // ignore j
113     if(toupper(msg[i])==toupper(msg[j])) // if duplicate add 'X' after the first letter
114     {
115         reqText[in]=toupper(msg[i]);
116         reqText[in+1]='X';
117         in++;
118     }
119     else
120     {
121         reqText[in]=toupper(msg[i]);
122     }
123     in++;
124 }
125 if(in%2!=0) reqText[in]='X'; // if one character Left, add 'X' after it
126
127 cout<<"Cypher text: ";
128 //encipher starts
129 int P,O,R,S,f1,f2;
```

Compiler Resources Compile Log Debug Find Results Console

Line: 100 Col: 37 Sel: 0 Lines: 232 Length: 5536 Insert Done parsing in 0.016 seconds

D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\5.Playfair.cpp - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C | 5.Playfair.cpp | 6.HillCipher.cpp | 7.RailFence.cpp | 8.RowTransposition.cpp | 11.StreamCipherTechnique.cpp

```
130 char x,y;
131 for(int i=0;i<in;i+=2)
132 {
133     x=reqText[i];
134     y=reqText[i+1];
135     f1=f2=0;
136     for(int j=0;j<5;j++)
137     {
138         for(int k=0;k<5;k++)
139         {
140             if(x==grid[j][k])
141             {
142                 P=j;
143                 Q=k;
144                 f1=1;
145             }
146             if(y==grid[j][k])
147             {
148                 R=j;
149                 S=k;
150                 f2=1;
151             }
152             if(f1 && f2) break;
153         }
154         if(f1 && f2) break;
155     }
156     if(P==R) //same row
157     {
158         if(Q==4) cout<<grid[P][0];
159         else cout<<grid[P][Q+1];
160         if(S==4) cout<<grid[R][0];
161         else cout<<grid[R][S+1];
162     }
```

Compiler Resources Compile Log Debug Find Results Console

Line: 100 Col: 37 Sel: 0 Lines: 232 Length: 5536 Insert Done parsing in 0.016 seconds

D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\5.Playfair.cpp - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C | 5.Playfair.cpp | 6.HillCipher.cpp | 7.RailFence.cpp | 8.RowTransposition.cpp | 11.StreamCipherTechnique.cpp

```
174         cout<<grid[P][S]<<grid[R][Q];
175     }
176     cout<<endl<<endl;
177     menu();
178 }
179 void createGrid()
180 {
181     cout<<"Keyword: ";
182     cin>>keyword; // key of the cypher
183     getChar();
184     len=strlen(keyword); // size of input string 0(n) :3
185     mark['I']=1; // ignore I
186     r=0,c=0; //initialize row and column
187     // first populate the keyword
188     for(int i=0;i<len;i++)
189     {
190         if(!mark[toupper(keyword[i])]) // ignore duplicates
191         {
192             mark[toupper(keyword[i])]=1;
193             grid[r][c]=toupper(keyword[i]);
194             if(c%5==0) //increase row column
195             {
196                 c=0;
197                 r++;
198             }
199         }
200     }
201     // complete rest of the matrix from unused characters starting from A
202     for(int i='A';i<='Z';i++)
203     {
```

Compiler Resources Compile Log Debug Find Results Console

Line: 100 Col: 37 Sel: 0 Lines: 232 Length: 5536 Insert Done parsing in 0.016 seconds

The screenshot shows a C++ IDE with the file `5.Playfair.cpp` open. The code implements a 5x5 matrix for a cipher. It starts by defining a 5x5 grid and a mark array. The `showGrid()` function prints the matrix. The main logic is in the `for(int i='A'; i<='Z'; i++)` loop, which populates the matrix with characters from 'A' to 'Z', skipping characters already in the matrix. The code is as follows:

```
201 // complete rest of the matrix from unused characters starting from A
202 for(int i='A'; i<='Z'; i++)
203 {
204     if(mark[i]==0) // taking values that are not in the matrix already
205     {
206         grid[r][c++] = i;
207         mark[i] = 1;
208         if(c%5==0)
209         {
210             if(r==4 && c==5) break; //matrix populated
211             // else increase row column
212             r++;
213             c=0;
214         }
215     }
216 }
217
218 void showGrid()
219 {
220     cout<<"5x5 Matrix"<<endl;
221     for(int i=0; i<5; i++)
222     {
223         for(int j=0; j<5; j++)
224         {
225             cout<<grid[i][j]<<" ";
226         }
227         cout<<endl;
228     }
229 }
```

The status bar at the bottom shows: Line: 204 Col: 6 Sel: 0 Lines: 232 Length: 5536 Insert Done parsing in 0.016 seconds.

OUTPUT

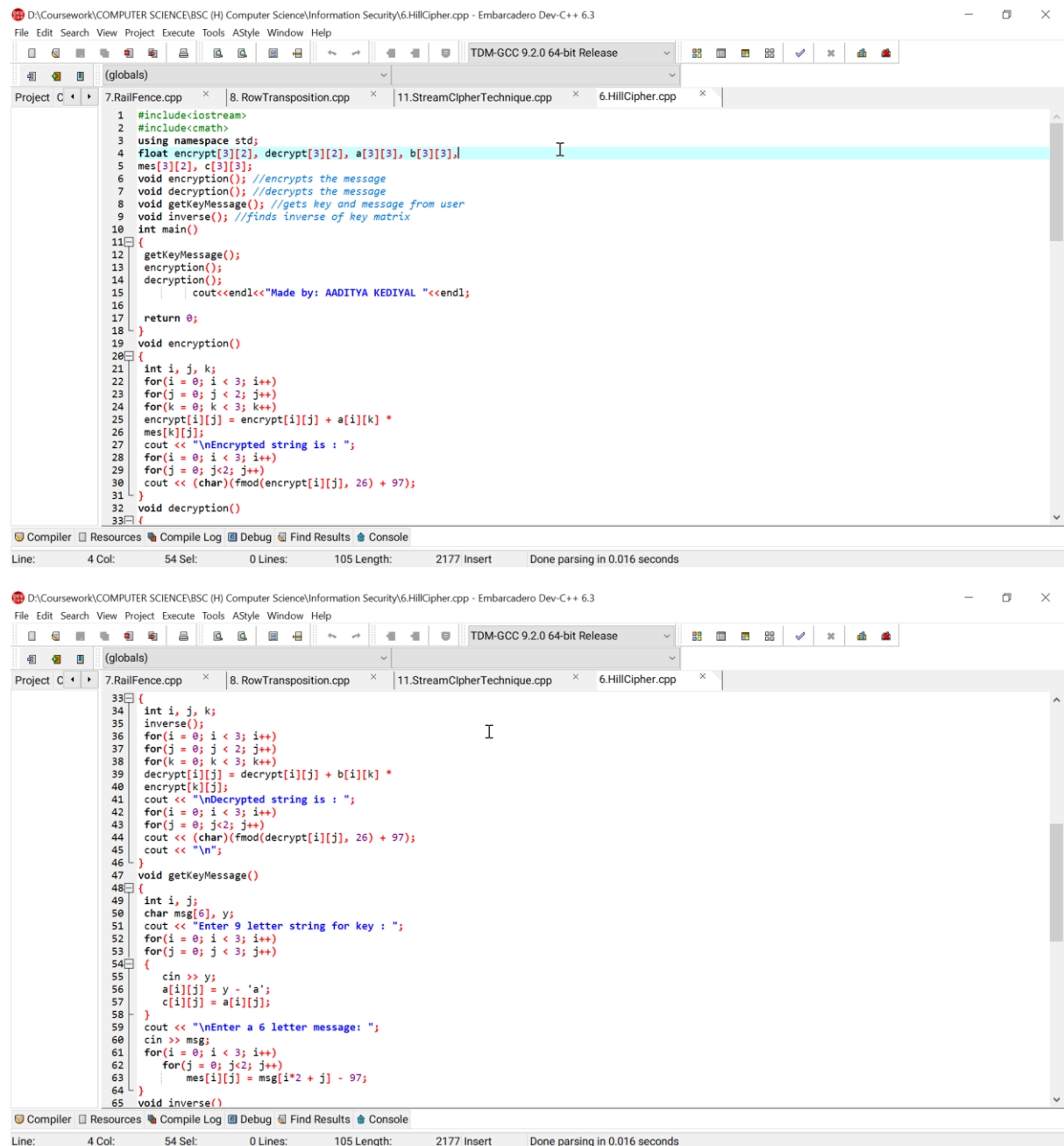
The screenshot shows the output of the `Playfair.exe` application. It displays the 5x5 matrix and the resulting cypher text for the message "iamaadi".

```
D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\5.Playfair.exe
3. Exit
Enter choice: 1
Keyword: aadi
5x5 Matrix
A D I B C
E F G H K
L M N O P
Q R S T U
V W X Y Z
Message to cypher: iamaadi
Cypher text: BDLDDVIIB

1. Encipher
2. Decipher
3. Exit
Enter choice: 2
Keyword: aadi
5x5 Matrix
A D I B C
E F G H K
L M N O P
Q R S T U
V W X Y Z
Cypher text: BDLDDVIIB
Decipher text: IAMAXADI

1. Encipher
2. Decipher
3. Exit
Enter choice:
```

6. Implement hill cipher substitution operation



```
1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4 float encrypt[3][2], decrypt[3][2], a[3][3], b[3][3];
5 mes[3][2], c[3][3];
6 void encryption(); //encrypts the message
7 void decryption(); //decrypts the message
8 void getKeyMessage(); //gets key and message from user
9 void inverse(); //finds inverse of key matrix
10 int main()
11 {
12     getKeyMessage();
13     encryption();
14     decryption();
15     cout<<endl<<"Made by: AADITYA KEDIYAL "<<endl;
16 }
17 return 0;
18 }
19 void encryption()
20 {
21     int i, j, k;
22     for(i = 0; i < 3; i++)
23     for(j = 0; j < 2; j++)
24     for(k = 0; k < 3; k++)
25         encrypt[i][j] = encrypt[i][j] + a[i][k] *
26         mes[k][j];
27     cout << "\nEncrypted string is : ";
28     for(i = 0; i < 3; i++)
29     for(j = 0; j < 2; j++)
30         cout << (char)(fmod(encrypt[i][j], 26) + 97);
31 }
32 void decryption()
33 {
34     int i, j, k;
35     inverse();
36     for(i = 0; i < 3; i++)
37     for(j = 0; j < 2; j++)
38     for(k = 0; k < 3; k++)
39         decrypt[i][j] = decrypt[i][j] + b[i][k] *
40         encrypt[k][j];
41     cout << "\nDecrypted string is : ";
42     for(i = 0; i < 3; i++)
43     for(j = 0; j < 2; j++)
44         cout << (char)(fmod(decrypt[i][j], 26) + 97);
45     cout << "\n";
46 }
47 void getKeyMessage()
48 {
49     int i, j;
50     char msg[6], y;
51     cout << "Enter 9 letter string for key : ";
52     for(i = 0; i < 3; i++)
53     for(j = 0; j < 3; j++)
54     {
55         cin >> y;
56         a[i][j] = y - 'a';
57         c[i][j] = a[i][j];
58     }
59     cout << "\nEnter a 6 letter message: ";
60     cin >> msg;
61     for(i = 0; i < 3; i++)
62     for(j = 0; j < 2; j++)
63         mes[i][j] = msg[i*2 + j] - 97;
64 }
65 void inverse()
```

```
72     if(i == j)
73         b[i][j] = 1;
74     else
75         b[i][j] = 0;
76 }
77 for(k = 0; k < 3; k++)
78 {
79     for(i = 0; i < 3; i++)
80     {
81         p = c[i][k];
82         q = c[k][k];
83         for(j = 0; j < 3; j++)
84         {
85             if(i != k)
86             {
87                 c[i][j] = c[i][j] * q - p * c[k][j];
88                 b[i][j] = b[i][j] * q - p * b[k][j];
89             }
90         }
91     }
92 }
93 for(i = 0; i < 3; i++)
94     for(j = 0; j < 3; j++)
95         b[i][j] = ceil((b[i][j] / c[i][i]) * 10000) / 10000.0;
96
97 cout << "\n\nInverse Matrix is:\n";
98 for(i = 0; i < 3; i++)
99 {
100     for(j = 0; j < 3; j++)
101         cout << b[i][j] << "\t";
102     cout << "\n";
103 }
104 }
```

Line: 4 Col: 54 Sel: 0 Lines: 105 Length: 2177 Insert Done parsing in 0.016 seconds

OUTPUT

```
D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\6.HillCipher.exe
Enter 9 letter string for key : abcdefghi
Enter a 6 letter message: ananya
Encrypted string is : wnqnkn

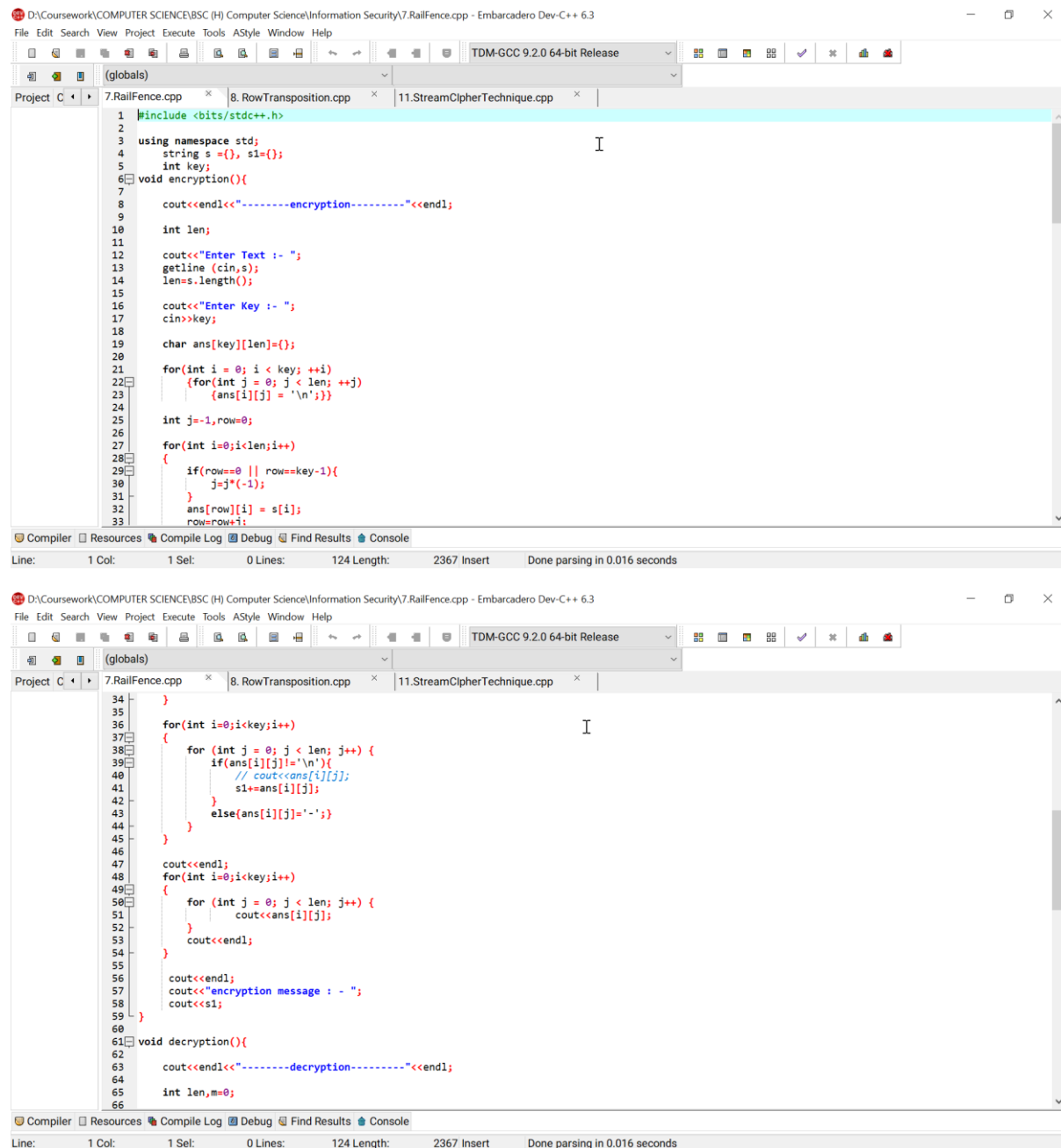
Inverse Matrix is:
nan nan nan
nan nan nan
nan nan nan

Decrypted string is :

Made by: AADITYA KEDIYAL

-----
Process exited after 29.22 seconds with return value 0
Press any key to continue . . .
```

7. Implement rail fence cipher transposition operation.



The image shows two screenshots of a C++ IDE (Embarcadero Dev-C++ 6.3) implementing the Rail Fence cipher transposition operation. The code is written in 7.RailFence.cpp.

Top Screenshot: Shows the beginning of the `void encryption()` function. It includes headers, uses the `std` namespace, and declares variables `s`, `s1`, `key`, and `len`. It prompts the user to enter text and a key, then initializes a 2D array `ans` of size `key` by `len`.

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4 string s = {}, s1 = {};
5 int key;
6 void encryption(){
7
8     cout<<endl<<"-----encryption-----"<<endl;
9
10    int len;
11
12    cout<<"Enter Text :- ";
13    getline (cin,s);
14    len=s.length();
15
16    cout<<"Enter Key :- ";
17    cin>>key;
18
19    char ans[key][len]={};
20
21    for(int i = 0; i < key; ++i)
22        for(int j = 0; j < len; ++j)
23            ans[i][j] = '\n';
24
25    int j=-1,row=0;
26
27    for(int i=0;i<len;i++)
28    {
29        if(row==0 || row==key-1){
30            j=j*(-1);
31        }
32        ans[row][i] = s[i];
33        row=row+1;
```

Bottom Screenshot: Shows the continuation of the `encryption()` function. It completes the filling of the `ans` array by iterating over the key and the length of the string. It then prints the encrypted message.

```
34    }
35
36    for(int i=0;i<key;i++)
37    {
38        for (int j = 0; j < len; j++) {
39            if(ans[i][j]!='\n'){
40                // cout<<ans[i][j];
41                s1+=ans[i][j];
42            }
43            else{ans[i][j]='-';}
44        }
45    }
46
47    cout<<endl;
48    for(int i=0;i<key;i++)
49    {
50        for (int j = 0; j < len; j++) {
51            cout<<ans[i][j];
52        }
53        cout<<endl;
54    }
55
56    cout<<endl;
57    cout<<"encryption message : - ";
58    cout<<s1;
59 }
60
61 void decryption(){
62
63     cout<<endl<<"-----decryption-----"<<endl;
64
65     int len,m=0;
66 }
```

D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\7.RailFence.cpp - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

Project C++ 7.RailFence.cpp 8.RowTransposition.cpp 11.StreamCipherTechnique.cpp

```
67 len=s1.length();
68
69 char ans[key][len]={};
70
71 for(int i = 0; i < key; ++i)
72     for(int j = 0; j < len; ++j)
73         ans[i][j] = '\n';
74
75 int j=-1,row=0;
76
77 for(int i=0;i<len;i++)
78 {
79     if(row==0 || row==key-1){
80         j=j*(-1);
81     }
82     ans[row][i] = '*';
83     row=row+j;
84 }
85
86 cout<<endl;
87 for(int i=0;i<key;i++)
88 {
89     for (int j = 0; j < len; j++) {
90         if(ans[i][j]=='*'){ans[i][j]=s1[m++];}
91     }
92 }
93
94 for(int i=0;i<key;i++)
95 {
96     for (int j = 0; j < len; j++) {
97         if(ans[i][j]=='\n'){cout<<"-";}
98         else {cout<<ans[i][j];}
99     }
```

Compiler Resources Compile Log Debug Find Results Console

Line: 1 Col: 1 Sel: 0 Lines: 124 Length: 2367 Insert Done parsing in 0.016 seconds

D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\7.RailFence.cpp - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

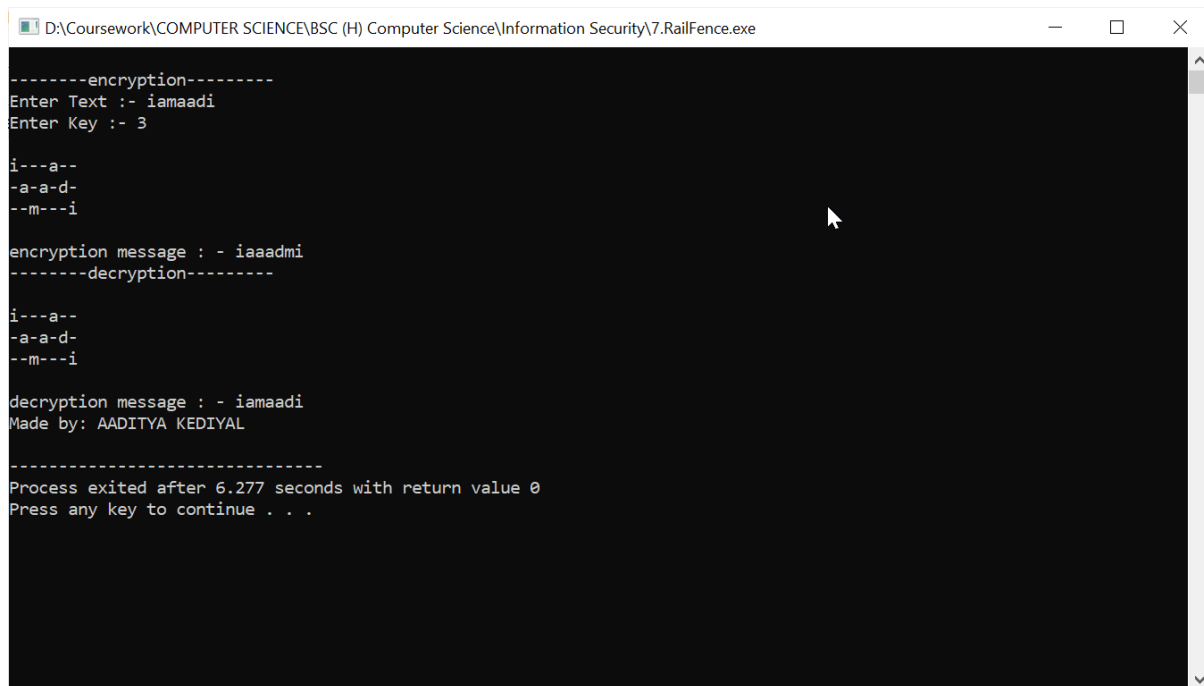
Project C++ 7.RailFence.cpp 8.RowTransposition.cpp 11.StreamCipherTechnique.cpp

```
93
94
95 for(int i=0;i<key;i++)
96 {
97     for (int j = 0; j < len; j++) {
98         if(ans[i][j]=='\n'){cout<<"-";}
99         else {cout<<ans[i][j];}
100     }
101     cout<<endl;
102 }
103
104 cout<<endl;
105 cout<<"decryption message : - ";
106 row =0, j=-1;
107 for(int i = 0; i < len; i++){
108     if(row == 0 || row == key-1)
109         {j= j * (-1);}
110     cout<<ans[row][i];
111     row = row + j;
112 }
113
114
115
116 int main()
117 {
118     encryption();
119     decryption();
120     cout<<endl<<"Made by: AADITYA KEDIYAL "<<endl;
121
122     return 0;
123 }
124
```

Compiler Resources Compile Log Debug Find Results Console

Line: 1 Col: 1 Sel: 0 Lines: 124 Length: 2367 Insert Done parsing in 0.016 seconds

OUTPUT



```
D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\7.RailFence.exe

-----encryption-----
Enter Text :- iamaadi
Enter Key :- 3

i---a--
-a-a-d-
--m---i

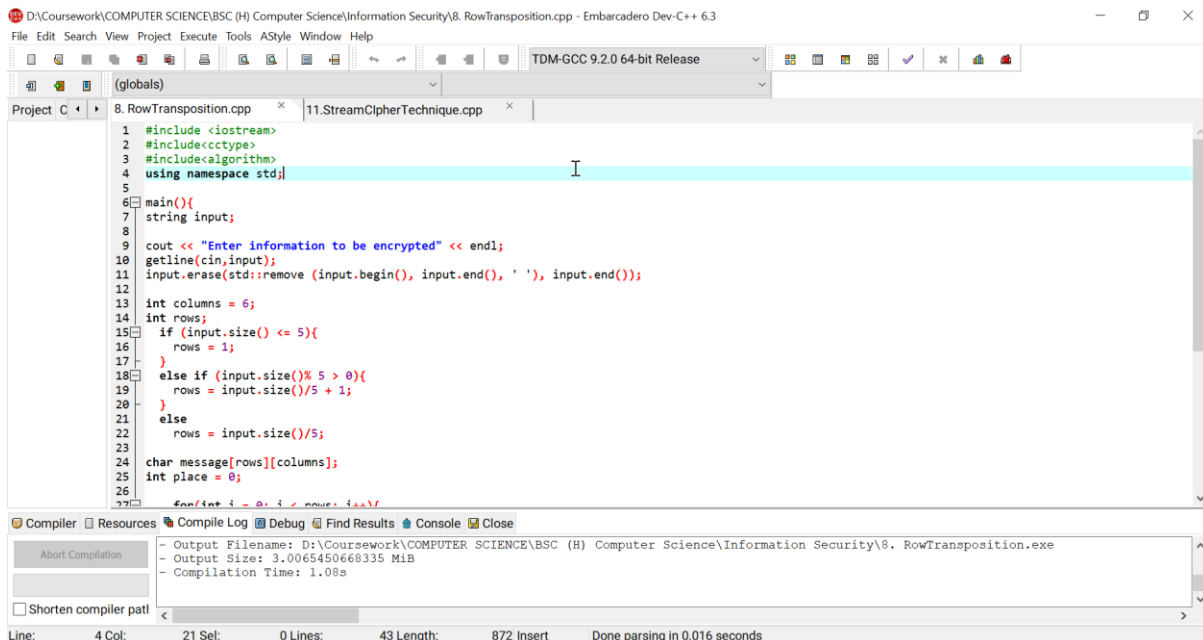
encryption message : - iaaadmi
-----decryption-----

i---a--
-a-a-d-
--m---i

decryption message : - iamaadi
Made by: AADITYA KEDIYAL

-----
Process exited after 6.277 seconds with return value 0
Press any key to continue . . .
```

8 Implement row transposition cipher transposition operation.



```
D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\8. RowTransposition.cpp - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

8. RowTransposition.cpp 11.StreamCipherTechnique.cpp
(globals)

1 #include <iostream>
2 #include <cctype>
3 #include <algorithm>
4 using namespace std;
5
6 main(){
7     string input;
8
9     cout << "Enter information to be encrypted" << endl;
10    getline(cin, input);
11    input.erase(std::remove(input.begin(), input.end(), ' '), input.end());
12
13    int columns = 6;
14    int rows;
15    if (input.size() <= 5){
16        rows = 1;
17    }
18    else if (input.size()% 5 > 0){
19        rows = input.size()/5 + 1;
20    }
21    else
22        rows = input.size()/5;
23
24    char message[rows][columns];
25    int place = 0;
26
27    for(int i = 0; i < rows; i++){
```

Compiler Resources Compile Log Debug Find Results Console Close

Abort Compilation

Shorten compiler path

Output Filename: D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\8. RowTransposition.exe
Output Size: 3.0065450668335 MiB
Compilation Time: 1.08s

Line: 4 Col: 21 Sel: 0 Lines: 43 Length: 872 Insert Done parsing in 0.016 seconds

The screenshot shows a C++ IDE with the following details:

- Title Bar:** D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\8. RowTransposition.cpp - Embarcadero Dev-C++ 6.3
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Includes icons for file operations, compilation, and debugging. The compiler is set to TDM-GCC 9.2.0 64-bit Release.
- Project Explorer:** Shows two files: 8. RowTransposition.cpp (selected) and 11.StreamCipherTechnique.cpp.
- Source Code (8. RowTransposition.cpp):**

```
18 else if (input.size()% 5 > 0){
19     rows = input.size()/5 + 1;
20 }
21 else
22     rows = input.size()/5;
23
24 char message[rows][columns];
25 int place = 0;
26
27 for(int i = 0; i < rows; i++){
28     for(int j = 0; j < (columns-1); j++){
29         if(place <= input.size()){
30             message[rows][columns] = input[place];
31         }
32         else {
33             message[rows][columns] = 'x';
34         }
35         place++;
36         message[rows][5] = 'y';
37         cout << message[rows][columns];
38     }
39     cout << endl;
40 }
41
42
43 }
```
- Compiler Output:**
 - Output Filename: D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\8. RowTransposition.exe
 - Output Size: 3.0065450668335 MiB
 - Compilation Time: 1.08s
- Status Bar:** Line: 20 Col: 4 Sel: 0 Lines: 43 Length: 872 Insert Done parsing in 0.016 seconds

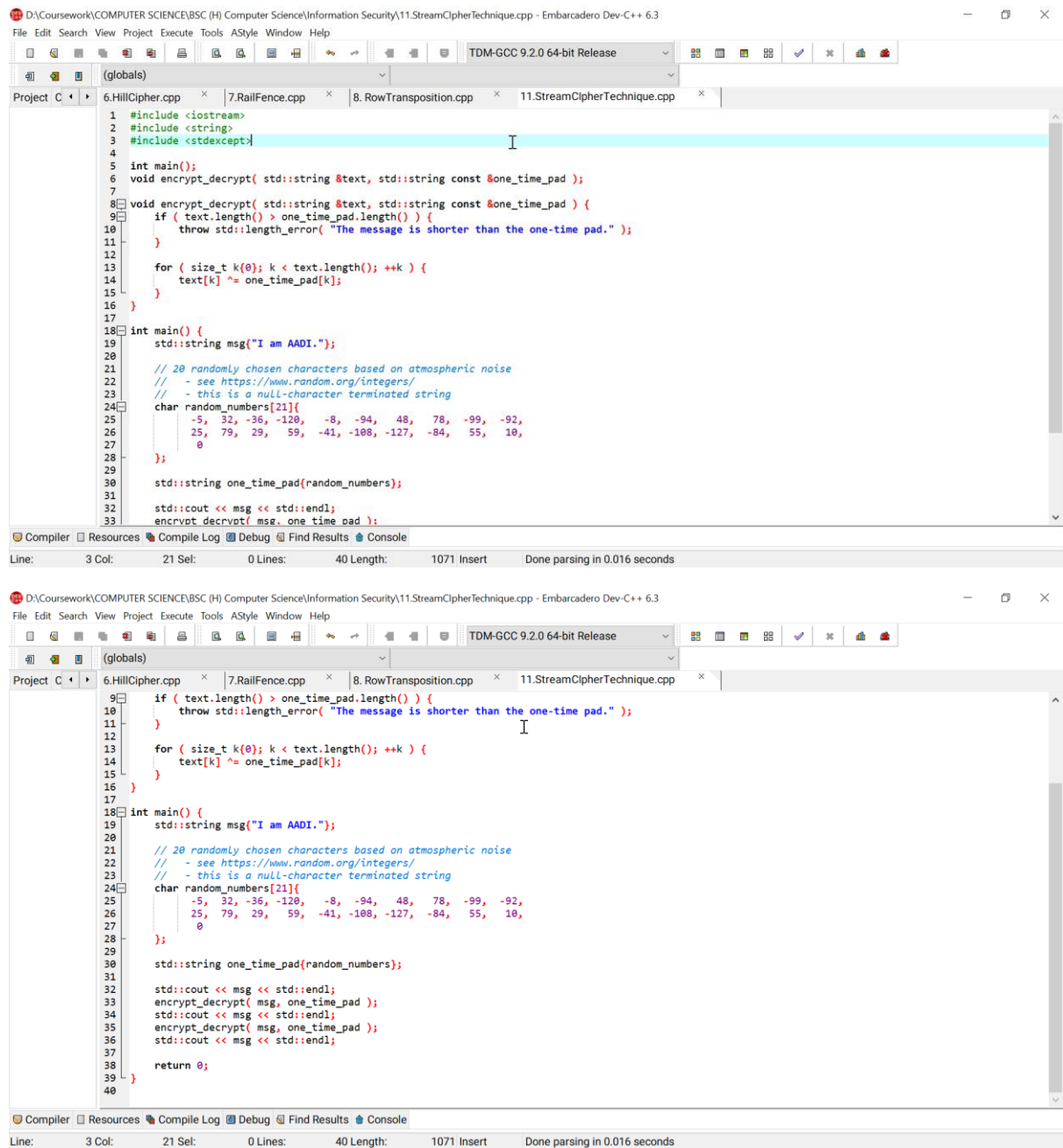
OUTPUT

The screenshot shows the execution of the RowTransposition.exe application with the following output:

```
Enter information to be encrypted
iamaadi
iamaa
di xx

-----
Process exited after 3.916 seconds with return value 0
Press any key to continue . . .
```

11. Implement a stream cipher technique



```
1 #include <iostream>
2 #include <string>
3 #include <stdexcept>
4
5 int main();
6 void encrypt_decrypt( std::string &text, std::string const &one_time_pad );
7
8 void encrypt_decrypt( std::string &text, std::string const &one_time_pad ) {
9     if ( text.length() > one_time_pad.length() ) {
10         throw std::length_error( "The message is shorter than the one-time pad." );
11     }
12
13     for ( size_t k(0); k < text.length(); ++k ) {
14         text[k] ^= one_time_pad[k];
15     }
16 }
17
18 int main() {
19     std::string msg("I am AADI.");
20
21     // 20 randomly chosen characters based on atmospheric noise
22     // - see https://www.random.org/integers/
23     // - this is a null-character terminated string
24     char random_numbers[21]{
25         -5, 32, -36, -120, -8, -94, 48, 78, -99, -92,
26         25, 79, 29, 59, -41, -108, -127, -84, 55, 10,
27         0
28     };
29
30     std::string one_time_pad(random_numbers);
31
32     std::cout << msg << std::endl;
33     encrypt_decrypt( msg, one_time_pad );
34     std::cout << msg << std::endl;
35     encrypt_decrypt( msg, one_time_pad );
36     std::cout << msg << std::endl;
37
38     return 0;
39 }
40
```

OUTPUT

```
D:\Coursework\COMPUTER SCIENCE\BSC (H) Computer Science\Information Security\11.StreamCipherTechnique.exe
I am AADI.
I am AADI.
I am AADI.
-----
Process exited after 0.0885 seconds with return value 0
Press any key to continue . . .
```