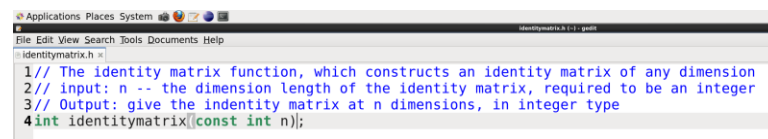


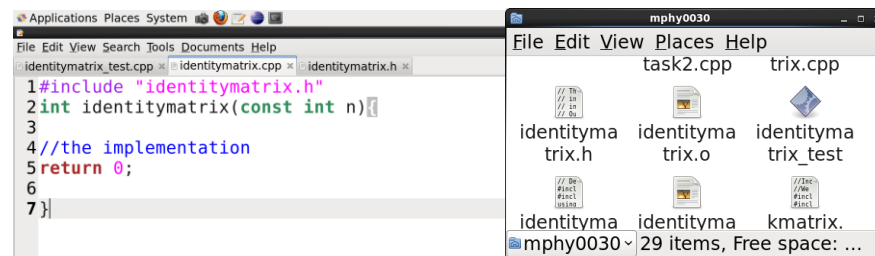
## Task 1

1.1: Create an “identitymatrix.h” function, the return type is expected to be an integer (int); function identifier identitymatrix. A constant integer n is expected in order to be consistent with output.



```
1// The identity matrix function, which constructs an identity matrix of any dimension
2// input: n -- the dimension length of the identity matrix, required to be an integer
3// Output: give the identity matrix at n dimensions, in integer type
4int identitymatrix(const int n);
```

1.2: Create source file and the implementation of the 1.1 function with no statement. Compile the “identitymatrix.cpp” file to produce the corresponding objective file (\*.o).

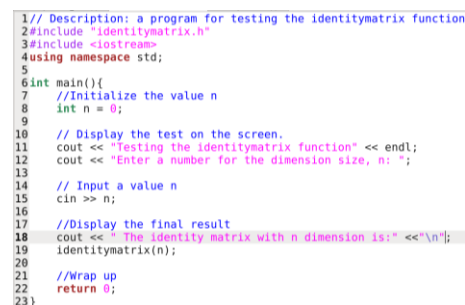


```
1#include "identitymatrix.h"
2int identitymatrix(const int n){
3
4//the implementation
5return 0;
6
7}
```

(code)

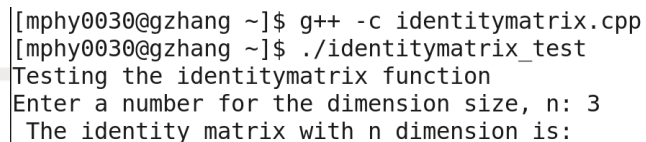
(Result)

1.3 : Create a “.cpp” test file for testing the identitymatrix function declared and implemented in 1.1 and 1.2. Compile this file and link it with the objective function from 1.2 to produce an executable for testing your function.



```
1// Description: a program for testing the identitymatrix function
2#include "identitymatrix.h"
3#include <iostream>
4using namespace std;
5
6int main(){
7    //Initialize the value n
8    int n = 0;
9
10   // Display the test on the screen.
11   cout << "Testing the identitymatrix function" << endl;
12   cout << "Enter a number for the dimension size, n: ";
13
14   // Input a value n
15   cin >> n;
16
17   //Display the final result
18   cout << "The identity matrix with n dimension is:" << endl;
19   identitymatrix(n);
20
21   //Wrap up
22   return 0;
23}
```

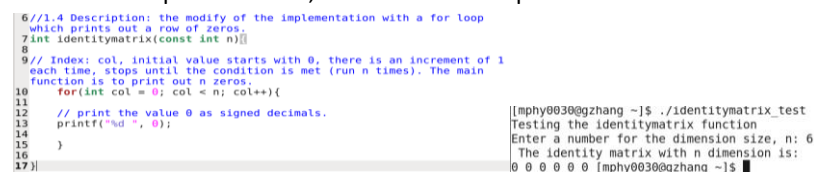
(code)



```
[mphy0030@gzhang ~]$ g++ -c identitymatrix.cpp
[mphy0030@gzhang ~]$ ./identitymatrix_test
Testing the identitymatrix function
Enter a number for the dimension size, n: 3
The identity matrix with n dimension is:
```

(Result)

1.4: Modify my implementation in 1.2 with a for loop which prints out a row of n zeros. Test the result of the new implementation, which works as expected.



```
6//1.4 Description: the modify of the implementation with a for loop
7// which prints out a row of zeros.
8int identitymatrix(const int n){
9
10 // Index: col, initial value starts with 0, there is an increment of 1
11 // each time, stops until the condition is met (run n times). The main
12 // function is to print out n zeros.
13 for(int col = 0; col < n; col++){
14     // print the value 0 as signed decimals.
15     printf("%d ", 0);
16 }
17}
```

```
[mphy0030@gzhang ~]$ ./identitymatrix_test
Testing the identitymatrix function
Enter a number for the dimension size, n: 6
The identity matrix with n dimension is:
0 0 0 0 0 0 [mphy0030@gzhang ~]$
```

(code)

(Result)

1.5 :Modify the implementation in 1.4 by writing an outer loop, the function of the outer loop is to print out n rows of zeros. Display the result with the new implementation for task 1.5 (Result).

```

6//1.5 Description: the modification of the implementation in 1.4, with
an outer loop which prints out n rows of zeros.
7int identitymatrix(const int n){
8
9// The main function containing inner and outer loops are to print out
n rows of n columns of zeros, as the number of rows is supposed to
equal to the number of columns
10// Index: row, initial value starts with 0, there is an increment of 1
each time, stops until the condition is met (run n rows)
11// Index: col, initial value starts with 0, there is an increment of 1
each time, stops until the condition is met (run n columns).
12
13    for(int row = 0; row < n; row++){
14
15        for(int col = 0; col < n; col++){
16
17            // print the value 0 as signed decimals.
18            printf("%2d ", 0);
19        }
20        printf("\n");
21    }
22    return 0;
23
24}

```

```

Testing the identitymatrix function
Enter a number for the dimension size, n: 8
The identity matrix with n dimension is:
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

```

(Result)

1.6: Modify the inner loop to include an if else conditional statement to print out “1” when a certain condition is met and “0” otherwise (code). Display the output (Result).

```

9 // 1.6 The main function prints out an identity matrix
10 // Index: col, initial value starts with 0, there is an
increment of 1 each time, stops until the condition is met (run n
rows).
11 // Index: col, initial value starts with 0, there is an
increment of 1 each time, stops until the condition is met (run n
columns).
12
13    for(int row = 0; row < n; row++){
14
15        for(int col = 0; col < n; col++){
16            // Check if row is equal to column, if it is, prints out 1
17            if(row == col){
18                printf("%2d ", 1);
19            }
20            // otherwise, print the value 0 as signed decimals.
21            else{
22                printf("%2d ", 0);
23            }
24            printf("\n");
25        }
26    }
27    return 0;
28}

```

```

Enter a number for the dimension size, n: 5
The identity matrix with n dimension is:
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

```

(code)

(Result)

## Tasks 2

2.1: Update on the header file to make it more versatile for all the matrices, and rename it with “matrix.h”

```

1// The matrix function, which constructs a matrix of any dimension, it could be either identity,K,C,T or B matrix.
2// input: n -- the dimension length of the matrix, required to be an integer
3// input: choice -- the selection of the matrix, required to be an integer
4// Output: give the identity matrix at n dimensions, in integer type
5int matrix(const int choice,const int n);

```

2.2: Implementation of all the matrix (case 1 to case 5), all the functions are arranged in cases statements based on int choice. Also, all the functions in the “identitymatrix.cpp” file are declared and implemented, and this file was renamed with a new name “matrix.cpp”. The “identitymatrix\_test.cpp” was rewritten and I rename it with “matrix\_test.cpp”. For C matrix, it follows the matrix construction style of assigning value -1 to the position on top-right and bottom-left corner of the matrix.

```

1//Include the headerfile "matrix.h", which was known as "identitymatrix.h";
2//We need to use printf()function, so there is a need to include <stdio>
3#include "matrix.h"
4#include <stdio>
5#include <iostream>
6using namespace std;
7//1.5 Description: the modification of the implementation in 1.4, with an outer loop which prints out n rows of zeros.
8// Index: row, initial value starts with 0, there is an increment of 1 each time, stops until the condition is met (run n rows).
9// Index: col, initial value starts with 0, there is an increment of 1 each time, stops until the condition is met (run n columns).
10int matrix(const int choice,const int n){
11    switch(choice){
12        case 1: {
13            cout << "K(" << n << ") = " << endl;
14            // 1.6 The following function prints out an identity matrix
15            for(int row = 0; row < n; row++){
16                for(int col = 0; col < n; col++){
17                    // Check if row is equal to column, if it is, prints out 1
18                    if(row == col){
19                        printf("%2d ", 1);
20                    }
21                    // otherwise, print the value 0 as signed decimals.
22                    else{
23                        printf("%2d ", 0);
24                    }
25                }
26                printf("\n");
27            }
28            return 0;
29        }
30    }
31}

```

```

31    case 2: {
32        cout << "K(" << n << ") = " << endl;
33        for(int row = 0; row < n; row++){
34            for(int col = 0; col < n; col++){
35                // Check if row is equal to column, if it is, prints out 2.
36                // Check if row+1 is equal to column, if it is, prints out -1.
37                if(row == col){
38                    printf("%2d ", 2);
39                }
40                else if(row+1 == col || row-1 == col){
41                    printf("%2d ", (-1));
42                }
43                // otherwise, print the value 0 as signed 2 decimals.
44                else{
45                    printf("%2d ", 0);
46                }
47            }
48            printf("\n");
49        }
50        return 0;
51    }

```

```

53 case 3: {
54     cout << "C" << n << " " << endl;
55     for(int row = 0; row < n; row++){
56         for(int col = 0; col < n; col++){
57             // Check if row and column are equal to 0 or n-1, if it is, prints out 1.
58             // Check if row is equal to column, if it is, prints out 2.
59             // Check if row+1 is equal to column, if it is, prints out -1.
60             if(row == col){
61                 printf("%2d ", 2);
62             }
63             else if(row == n-1 && col == 0){
64                 printf("%2d ", -1);
65             }
66             else if(row == 0 && col == n-1){
67                 printf("%2d ", -1);
68             }
69             else if(row+1 == col || row-1 == col){
70                 printf("%2d ", -1);
71             }
72             // otherwise, print the value 0 as signed 2 decimals.
73             else{
74                 printf("%2d ", 0);
75             }
76             printf("\n");
77         }
78     }
79     return 0;
80 }

109 case 5: {
110     cout << "B" << n << " " << endl;
111     for(int row = 0; row < n; row++){
112         for(int col = 0; col < n; col++){
113             // Check if row and column are equal to 0 or n-1, if it is, prints out 1.
114             // Check if row is equal to column, if it is, prints out 2.
115             // Check if row+1 is equal to column, if it is, prints out -1.
116             if(row == col && row > 0 && col > 0 && row < n-1 && col < n-1){
117                 printf("%2d ", 2);
118             }
119             else if(row == 0 && col == 0){
120                 printf("%2d ", 1);
121             }
122             else if(row == n-1 && col == n-1){
123                 printf("%2d ", 1);
124             }
125             else if(row+1 == col || row-1 == col){
126                 printf("%2d ", -1);
127             }
128             // otherwise, print the value 0 as signed 2 decimals.
129             else{
130                 printf("%2d ", 0);
131             }
132             printf("\n");
133         }
134     }
135     return 0;
136 }
137 }
138 }

82 case 4: {
83     cout << "T" << n << " " << endl;
84     for(int row = 0; row < n; row++){
85         for(int col = 0; col < n; col++){
86             // Check if row is equal to column(except row=0 && column=0), if it is, prints out 2.
87             // Check if the function gives an output of 1 at row = 0 and col = 0.
88             // Check if row+1 is equal to column, if it is, prints out -1.
89             if(row == col && row > 0 && col > 0){
90                 printf("%2d ", 2);
91             }
92             else if(row == 0 && col == 0){
93                 printf("%2d ", 1);
94             }
95             else if(row+1 == col || row-1 == col){
96                 printf("%2d ", -1);
97             }
98             // otherwise, print the value 0 as signed 2 decimals.
99             else{
100                 printf("%2d ", 0);
101             }
102             printf("\n");
103         }
104     }
105     return 0;
106 }

```

### Task 3

3.1 and 3.2: Declare and implement a function prints out an introduction; Declare and implement a function prints out an introduction

```

3// Create a selection interface
4void introduction(){
5    cout << "Special Matrix Maker (R)" << endl;
6    cout << "Copyright Nov 2018 [Zihao Fang]" << endl;
7    cout << "Student number: 14018713" << endl;
8    cout << "This tool gives the user an opportunity to create a matrix of either identity matrix or K matrix or C matrix or T matrix, by entering the corresponding number 1-5. Otherwise, exit the tool (option 0)." << endl;
9}
10
11void selection(){
12    cout << "Choose from the following options: " << endl;
13    cout << "1 -> To create a I matrix; Enter '1' " << endl;
14    cout << "2 -> To create a K matrix; Enter '2' " << endl;
15    cout << "3 -> To create a C matrix; Enter '3' " << endl;
16    cout << "4 -> To create a T matrix; Enter '4' " << endl;
17    cout << "5 -> To create a B matrix; Enter '5' " << endl;
18    cout << "0 -> To Exit; Enter '0' " << endl;
19}
20
21int main(){
22    introduction();
23    selection();
24}

```

3.3: In task 2.2, all the matrix functions were dispatched in the switch statement in the case order 1-5

3.4: Implemented the matrix\_test.cpp script to create a tool for creating required matrices with n dimensions

```

1// Description: a program for creating the matrix function for identity,K,C,T and B matrices with n dimensions
2#include "matrix.h"
3#include <iostream>
4using namespace std;

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80

int main(){
    //Initialize the value n and the choice
    int n = 0;
    int choice = 0;
    introduction();
    selection();
    // Display the test on the screen.
    // Input a value choice
    cin >> choice;
    if(choice < 0 || choice > 5){
        cout << "Wrong number! Try 0-5, please!" << endl;
        cout << "Enter a number for the matrix selection option, n: ";
        cin >> choice;
    }
    // If a selection number is in the required range, 1-5, execute the matrix creation;
    // Only a positive n is allowed in this case, otherwise, return to the beginning input stage
    else if(choice != 0){
        cout << "Enter a number for the dimension size, n: ";
        // Input a value n
        cin >> n;
        while(n <= 0){
            cout << "A positive number of n is required!" << endl;
            cout << "Enter a number for the dimension size, n: ";
            // Input a value n
            cin >> n;
        }
        //Display the final result
        cout << "The matrix with " << n << " dimension is:" << endl;
        matrix(choice, n);
        selection();
        // Display the test on the screen.
        cout << "Enter a number for the matrix selection option, n: ";
        // Input a value choice
        cin >> choice;
    }
    //Wrap up
    return 0;
}

```

Result of Task2 and Task 3: Cases for selection number 0-5 with n=4;

<p>&lt;Special Matrix Maker (R)&gt;          Copyright Nov 2018 [Zihao Fang]          Student number: 14018713</p> <p>This tool gives the user an opportunity to create a matrix of either identity matrix or K matrix or C matrix or T matrix, by entering the corresponding number 1-5. Otherwise, exit the tool (option 0).</p> <p>Choose from the following options:          =&gt; To create a I matrix: Enter "1"          =&gt; To create a K matrix: Enter "2"          =&gt; To create a C matrix: Enter "3"          =&gt; To create a T matrix: Enter "4"          =&gt; To create a B matrix: Enter "5"          =&gt; To Exit: Enter "0"</p> <p>Enter a number for the matrix selection option, n: 1          Enter a number for the dimension size, n: 4</p>	<p>The matrix with 4 dimension is:</p> <p>I(4) =</p> <pre>1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1</pre> <p>Choose from the following options:          =&gt; To create a I matrix: Enter "1"          =&gt; To create a K matrix: Enter "2"          =&gt; To create a C matrix: Enter "3"          =&gt; To create a T matrix: Enter "4"          =&gt; To create a B matrix: Enter "5"          =&gt; To Exit: Enter "0"</p> <p>Enter a number for the matrix selection option, n: 2          Enter a number for the dimension size, n: 4</p>
<p>The matrix with 4 dimension is:</p> <p>K(4) =</p> <pre>2 -1 0 0 -1 2 -1 0 0 -1 2 -1 0 0 -1 2</pre> <p>Choose from the following options:          =&gt; To create a I matrix: Enter "1"          =&gt; To create a K matrix: Enter "2"          =&gt; To create a C matrix: Enter "3"          =&gt; To create a T matrix: Enter "4"          =&gt; To create a B matrix: Enter "5"          =&gt; To Exit: Enter "0"</p> <p>Enter a number for the matrix selection option, n: 3          Enter a number for the dimension size, n: 4</p>	<p>The matrix with 4 dimension is:</p> <p>C(4) =</p> <pre>2 -1 0 -1 -1 2 -1 0 0 -1 2 -1 -1 0 -1 2</pre> <p>Choose from the following options:          =&gt; To create a I matrix: Enter "1"          =&gt; To create a K matrix: Enter "2"          =&gt; To create a C matrix: Enter "3"          =&gt; To create a T matrix: Enter "4"          =&gt; To create a B matrix: Enter "5"          =&gt; To Exit: Enter "0"</p> <p>Enter a number for the matrix selection option, n: 4          Enter a number for the dimension size, n: 4</p>
<p>The matrix with 4 dimension is:</p> <p>T(4) =</p> <pre>1 -1 0 0 -1 2 -1 0 0 -1 2 -1 0 0 -1 2</pre> <p>Choose from the following options:          =&gt; To create a I matrix: Enter "1"          =&gt; To create a K matrix: Enter "2"          =&gt; To create a C matrix: Enter "3"          =&gt; To create a T matrix: Enter "4"          =&gt; To create a B matrix: Enter "5"          =&gt; To Exit: Enter "0"</p> <p>Enter a number for the matrix selection option, n: 5          Enter a number for the dimension size, n: 4</p>	<p>The matrix with 4 dimension is:</p> <p>B(4) =</p> <pre>1 -1 0 0 -1 2 -1 0 0 -1 2 -1 0 0 -1 1</pre> <p>Choose from the following options:          =&gt; To create a I matrix: Enter "1"          =&gt; To create a K matrix: Enter "2"          =&gt; To create a C matrix: Enter "3"          =&gt; To create a T matrix: Enter "4"          =&gt; To create a B matrix: Enter "5"          =&gt; To Exit: Enter "0"</p> <p>Enter a number for the matrix selection option, n: 0          Exit!</p>

Exceptions: where input choice numbers and n numbers are out of range 0-5.

```
Enter a number for the matrix selection option, n: 9
Wrong number! Try 0-5, please!
Enter a number for the matrix selection option, n: -1
Wrong number! Try 0-5, please!
Enter a number for the matrix selection option, n: 1
Enter a number for the dimension size, n: -1
A positive number of n is required!
Enter a number for the dimension size, n: 0
A positive number of n is required!
```