CPT111 PRINCIPLES OF PROGRAMMING

Pusat Pengajian Sains Komputer

Universiti Sains Malaysia

Semester I 2020/2021

Assignment : 1

Report for : Hackathon 1 Part A

Group Number : 44

Member List :

I. YEO YING SHENG (157267)
II. CHALLVEN JAPIRIN (157698)
III. EDU SINUSI (159136)

Lecturer's Name : Dr Nur Hana Samsudin

TABLE OF CONTENT

1.0) Problem Analysis

Develop a C++ programme to help a family choose the most appropriate dining table for family dining hall or dining room. The space at the dining hall must be optimised without over-crowding it. The program should prompt the dimension of the dining hall in length and width. The program will choose the comfortable space that has 1m space for the people's movement and chair usage and another 0.6m (minimum) to provide some space so there are some place to roam. The system should accept the length and width of the room as the user input. The program will determine the suitable table to choose based on the space requirement. The program should switch the value of room length and room width if the room width exceeds the room length. Apart from that, the program also does a comparison between the table so that it will give the best table that will provide a bigger space in a room. The program will show the maximum seater based on the table chosen. If any of the user input length and width of a room that does not exceed the minimum requirement of space, none of the table will be choose by the program.

2.0) Specification requirement:

2.1) Input:

1. Room length(meter)

2. Room width(meter)

2.2) Process:

1. Determine whether the room length is shorter than room width (roomLen < roomWid)

2. Calculate the space = 2 * (0.6 + 1)

3. Determine which suitable table (roomLen >= (3 + space) && roomWid >= (1 +space)) / (roomLen >= (3 + space) && roomWid >= (1.3 + space))/ (roomLen >= (2.1 + space) && roomWid >= (1 + space))/ (roomLen >= (1.5 + space) && roomWid >= (1.5 + space))/ (roomLen >= (1.35 + space) && roomWid >= (1.35 + space)) / (roomLen >= (1.5 + space) && roomWid >= (0.9 + space)) / (roomLen >= (1.3 + space) && roomWid >= (0.8 + space)) / (roomLen >= (0.9 + space) && roomWid >= (0.9 + space)) / (roomLen >= (0.76 + space) && roomWid >= (0.76 + space))

4. Determine whether the room length and width exceed the space of a table and minimum space requirement or not

2.3) Output:

1. Recommended table or tables.

2. Maximum seaters available.

3.0) Variables Definition:

Variables	Definition
roomLen	Length of room
roomWid	Width of room
cnt	Counter
space	Extra space required by each table
dimTmp	Temporary dimension

4.0) Design of Solution:

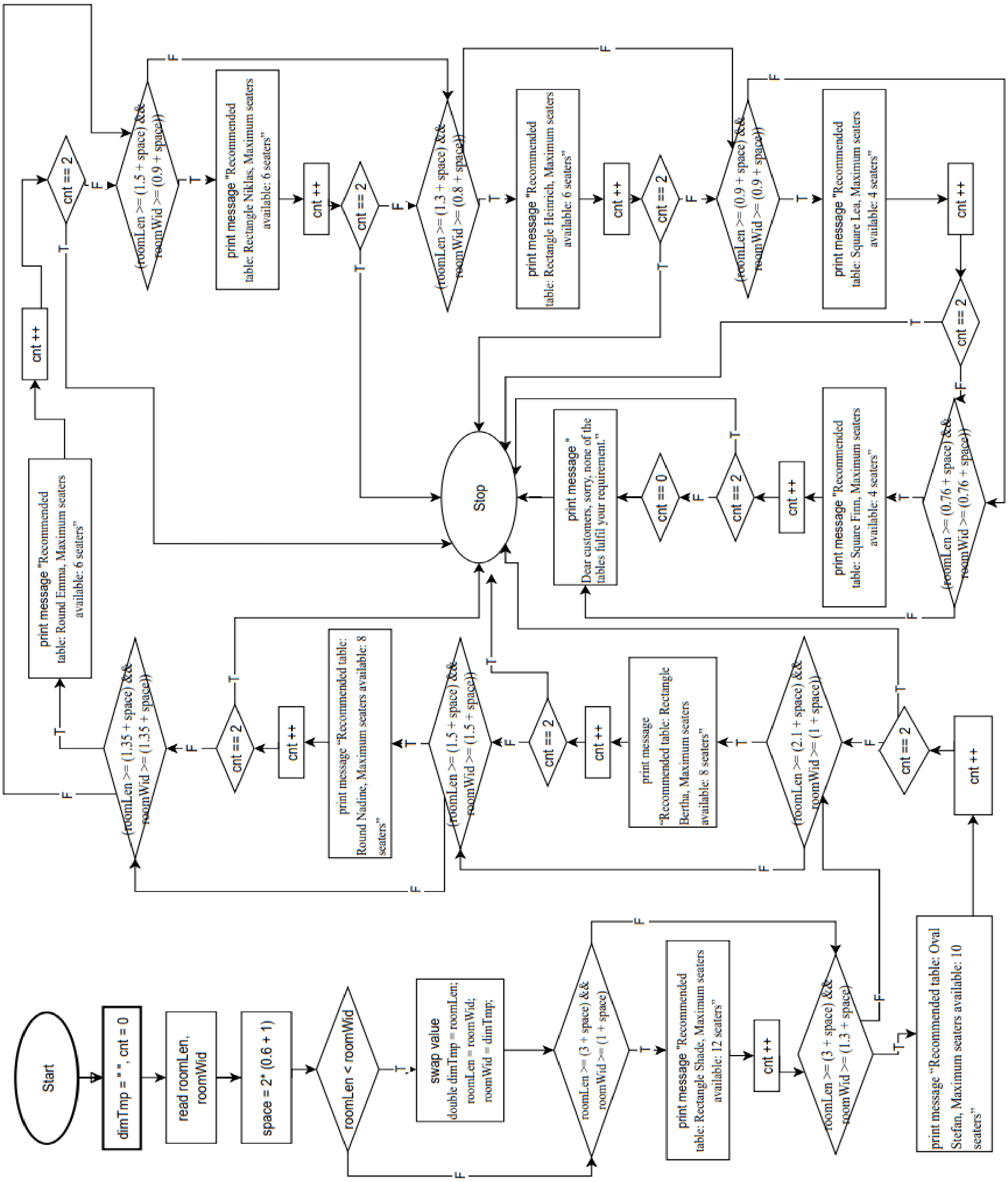
4.1) Pseudocode

```
Start
1.0  initialise dimTmp =  " ", cnt = 0,
2.0  read roomLen, roomWid
3.0  if (roomLen < roomWid)
        3.1  Swap the value of roomWid with roomLen if the condition is true
                3.1.1  dimTmp = roomLen
                3.1.2  roomLen = roomWid
                3.1.3  roomWid = dimTmp
4.0  Finding the minimum measurement of the additional space for people to roam and also for chair usage
        4.1  space = 2 * (0.6 + 1)
5.0  Find the most suitable table by doing comparison using if statements
      5.1  if (roomLen >= (3 + space) && roomWid >= (1 + space))
            5.1.2  print message "Recommended table: Rectangle Shade
                                        Maximum seaters available: 12 seaters"
                  5.1.3  cnt ++ 1
end if
   5.2  if (roomLen >= (3 + space) && roomWid >= (1.3 + space))
            5.2.1  print message "Recommended table: Oval Stefan
                                         Maximum seaters available: 10 seaters"
                             5.2.2  cnt ++ 1
                                   5.2.3  if (cnt == 2)
                end if
end if
```

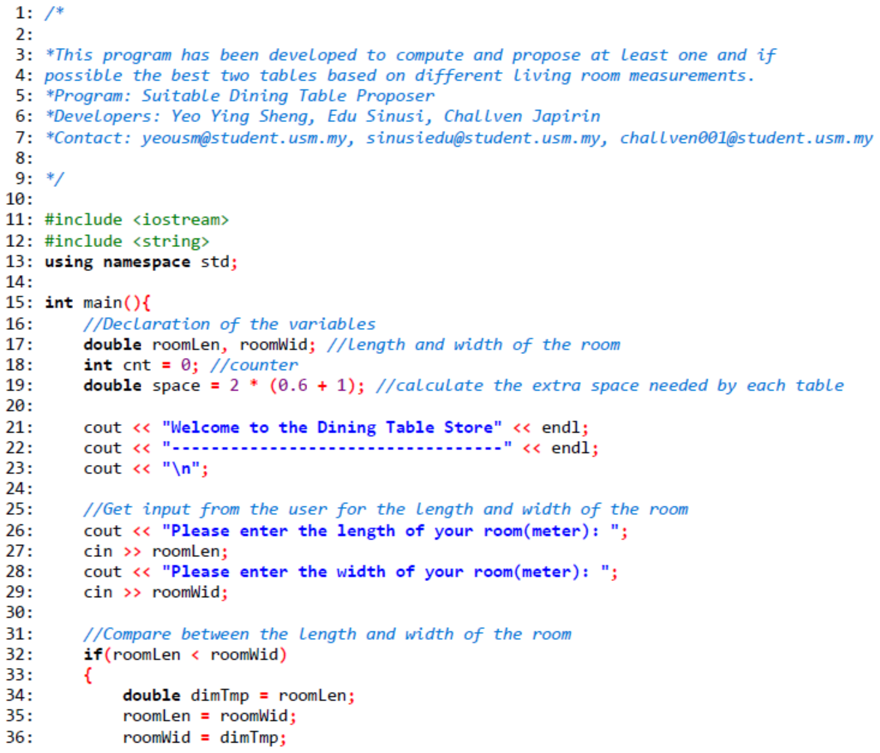
```
5.3  if (roomLen >= (2.1 + space) && roomWid >= (1 + space))
    5.3.1  print message "Recommended table: Rectangle Bertha
                 Maximum seaters available: 8 seaters"
        5.3.2  cnt ++ 1
            5.3.3  if (cnt == 2)
     end if
end if
5.4  if (roomLen >= (1.5 + space) && roomWid >= (1.5 + space))
    5.4.1  print message "Recommended table: Round Nadine
                Maximum seaters available: 8 seaters"
            5.4.2  cnt ++ 1
                5.4.3  If (cnt == 2)
     end if
end if
5.5  if (roomLen >= (1.35 + space) && roomWid >= (1.35 + space))
    5.5.1  print message "Recommended table: Round Emma
                 Maximum seaters available: 6 seaters"
           5.5.2  cnt ++ 1
                5.5.3  if (cnt == 2)
       end if
end if
5.6  if (roomLen >= (1.5 + space) && roomWid >= (0.9 + space))
   5.6.1  print message "Recommended table: Rectangle Emma
                Maximum seaters available: 6 seaters"
           5.6.2  cnt ++ 1
              5.6.3  if (cnt == 2)
     end if
end if
```

```
5.7  if (roomLen >= (1.3 + space) && roomWid >= (0.8 + space))
            5.7.1  print message "Recommended table: Rectangle Heinrich
                                         Maximum seaters available: 6 seaters"
                         5.7.2  cnt ++ 1
                                     5.7.3  if (cnt == 2)
              end if
end if
5.8  if (roomLen >= (0.9 + space) && roomWid >= (0.9 + space))
        5.8.1  print message "Recommended table: Square Lea
                                     Maximum seaters available: 4 seaters"
                   5.8.2  cnt ++ 1
                           5.8.3  if (cnt == 2)
              end if
end if
5.9  if (roomLen >= (0.76 + space) && roomWid >= (0.76 + space))
      5.9.1  print message "Recommended table: Square Finn
                                 Maximum seaters available: 2 seaters"
                 5.9.2  cnt ++ 1
                         5.9.3  if (cnt == 2)
              end if
end if
5.10  if (cnt == 0)
5.10.1  print message "Dear customers, sorry, none of the tables fulfil your requirement."
end if
Stop
```

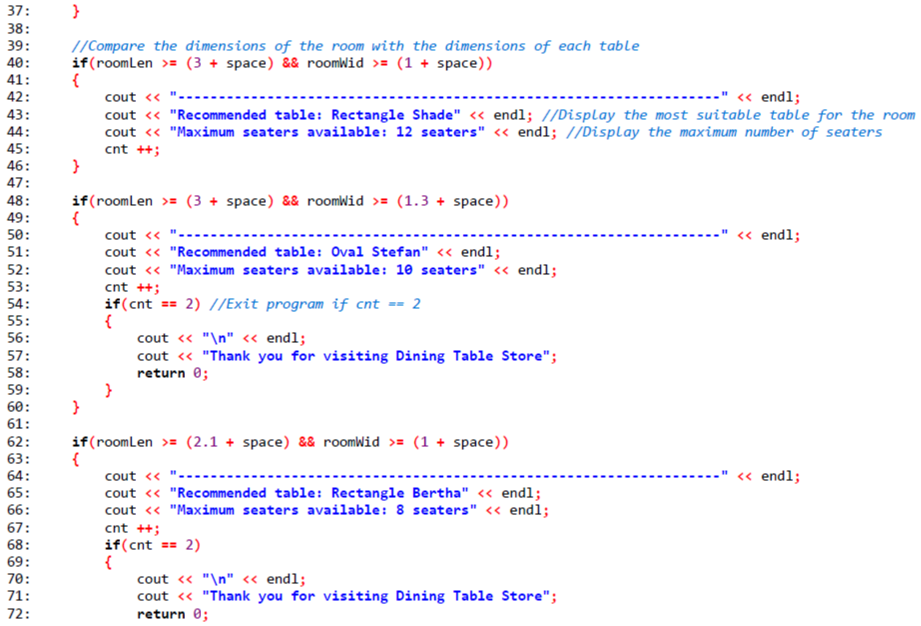
4.2) Flowchart


Start
dimTmp = " ", cnt = 0
read roomLen, roomWid
space = 2* (0.6 + 1)
roomLen < roomWid
T
F
swap value
double dimTmp = roomLen;
roomLen = roomWid;
roomWid = dimTmp;
roomLen >= (3 + space) &&
roomWid >= (1 + space)
T
F
print message "Recommended
table: Rectangle Shade, Maximum seaters
available: 12 seaters"
cnt ++
(roomLen >= (3 + space) &&
roomWid >= (1.3 + space))
T
F
print message "Recommended table: Oval
Stefan, Maximum seaters available: 10
seaters"
cnt ++
cnt == 2
T
F
(roomLen >= (2.1 + space) &&
roomWid >= (1 + space))
T
F
print message
"Recommended table: Rectangle
Bertha, Maximum seaters
available: 8 seaters"
cnt ++
cnt == 2
T
F
(roomLen >= (1.5 + space) &&
roomWid >= (1.5 + space))
T
F
print message "Recommended table:
Round Nadine, Maximum seaters available: 8
seaters"
cnt ++
cnt == 2
T
F
(roomLen >= (1.35 + space) &&
roomWid >= (1.35 + space))
T
F
print message "Recommended
table: Round Emma, Maximum seaters
available: 6 seaters"
cnt ++
cnt == 2
T
F
(roomLen >= (1.5 + space) &&
roomWid >= (0.9 + space))
T
F
print message "Recommended
table: Rectangle Niklas, Maximum seaters
available: 6 seaters"
cnt ++
cnt == 2
T
F
(roomLen >= (1.3 + space) &&
roomWid >= (0.8 + space))
T
F
print message "Recommended
table: Rectangle Heinrich, Maximum seaters
available: 6 seaters"
cnt ++
cnt == 2
T
F
(roomLen >= (0.9 + space) &&
roomWid >= (0.9 + space))
T
F
print message "Recommended
table: Square Lea, Maximum seaters
available: 4 seaters"
cnt ++
cnt == 2
T
F
(roomLen >= (0.76 + space) &&
roomWid >= (0.76 + space))
T
F
print message "Recommended
table: Square Finn, Maximum seaters
available: 4 seaters"
cnt ++
cnt == 2
T
F
cnt == 0
print message "
Dear customers, sorry, none of the
tables fulfil your requirement."
Stop

5.0) Source Code



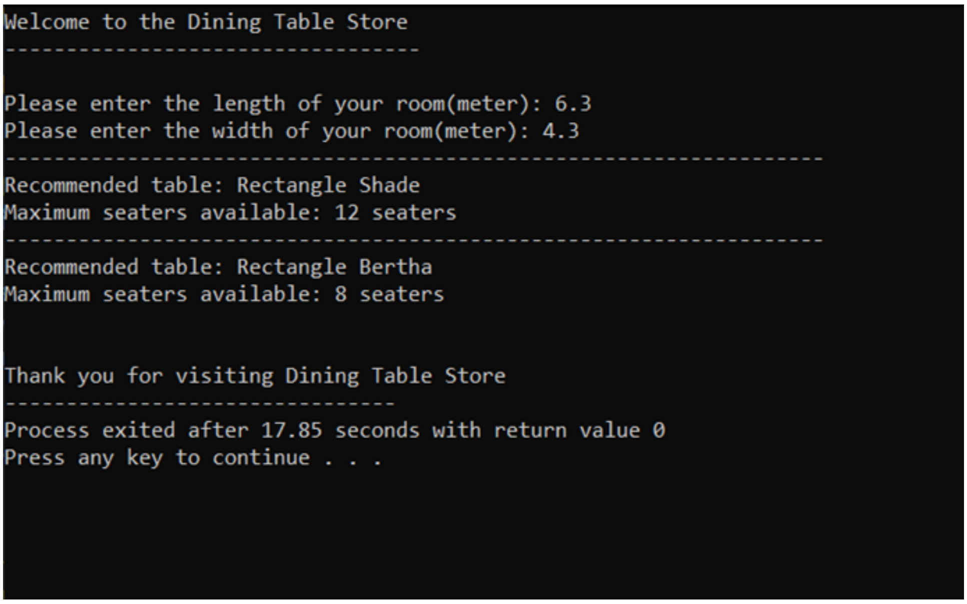
```cpp
/*

*This program has been developed to compute and propose at least one and if
possible the best two tables based on different living room measurements.
*Program: Suitable Dining Table Proposer
*Developers: Yeo Ying Sheng, Edu Sinusi, Challven Japirin
*Contact: yeousm@student.usm.my, sinusiedu@student.usm.my, challven001@student.usm.my

*/

#include <iostream>
#include <string>
using namespace std;

int main(){
    //Declaration of the variables
    double roomLen, roomWid; //length and width of the room
    int cnt = 0; //counter
    double space = 2 * (0.6 + 1); //calculate the extra space needed by each table

    cout << "Welcome to the Dining Table Store" << endl;
    cout << "---------------------------------" << endl;
    cout << "\n";

    //Get input from the user for the length and width of the room
    cout << "Please enter the length of your room(meter): ";
    cin >> roomLen;
    cout << "Please enter the width of your room(meter): ";
    cin >> roomWid;

    //Compare between the length and width of the room
    if(roomLen < roomWid)
    {
        double dimTmp = roomLen;
        roomLen = roomWid;
        roomWid = dimTmp;
```



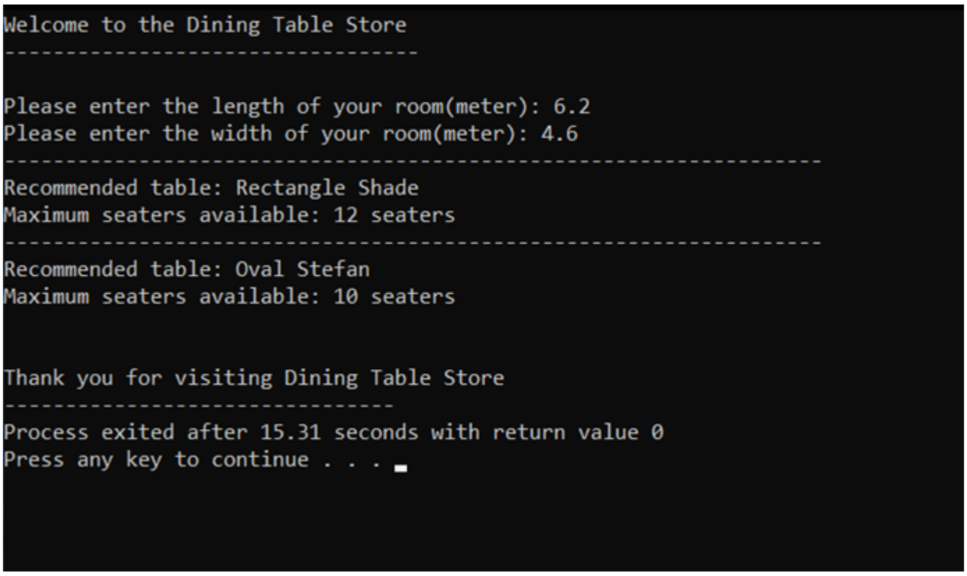
```cpp
    }

    //Compare the dimensions of the room with the dimensions of each table
    if(roomLen >= (3 + space) && roomWid >= (1 + space))
    {
        cout << "--------------------------------------------------------------------" << endl;
        cout << "Recommended table: Rectangle Shade" << endl; //Display the most suitable table for the room
        cout << "Maximum seaters available: 12 seaters" << endl; //Display the maximum number of seaters
        cnt ++;
    }

    if(roomLen >= (3 + space) && roomWid >= (1.3 + space))
    {
        cout << "--------------------------------------------------------------------" << endl;
        cout << "Recommended table: Oval Stefan" << endl;
        cout << "Maximum seaters available: 10 seaters" << endl;
        cnt ++;
        if(cnt == 2) //Exit program if cnt == 2
        {
            cout << "\n" << endl;
            cout << "Thank you for visiting Dining Table Store";
            return 0;
        }
    }

    if(roomLen >= (2.1 + space) && roomWid >= (1 + space))
    {
        cout << "--------------------------------------------------------------------" << endl;
        cout << "Recommended table: Rectangle Bertha" << endl;
        cout << "Maximum seaters available: 8 seaters" << endl;
        cnt ++;
        if(cnt == 2)
        {
            cout << "\n" << endl;
            cout << "Thank you for visiting Dining Table Store";
            return 0;
        }
    }

    if(roomLen >= (1.5 + space) && roomWid >= (1.5 + space))
    {
        cout << "-------------------------------------------------------------------" << endl;
        cout << "Recommended table: Round Nadine" << endl;
        cout << "Maximum seaters available: 8 seaters" << endl;
        cnt ++;
        if(cnt == 2)
        {
            cout << "\n" << endl;
            cout << "Thank you for visiting Dining Table Store";
            return 0;
        }
    }

    if(roomLen >= (1.35 + space) && roomWid >= (1.35 + space))
    {
        cout << "-------------------------------------------------------------------" << endl;
        cout << "Recommended table: Round Emma" << endl;
        cout << "Maximum seaters available: 6 seaters" << endl;
        cnt ++;
        if(cnt == 2)
        {
            cout << "\n" << endl;
            cout << "Thank you for visiting Dining Table Store";
            return 0;
        }
    }

    if(roomLen >= (1.5 + space) && roomWid >= (0.9 + space))
    {
        cout << "-------------------------------------------------------------------" << endl;
        cout << "Recommended table: Rectangle Niklas" << endl;
        cout << "Maximum seaters available: 6 seaters" << endl;
        cnt ++;
        if(cnt == 2)
        {
            cout << "\n" << endl;
            cout << "Thank you for visiting Dining Table Store";
            return 0;
        }
    }

    if(roomLen >= (1.3 + space) && roomWid >= (0.8 + space))
    {
        cout << "-------------------------------------------------------------------" << endl;
        cout << "Recommended table: Rectangle Heinrich" << endl;
        cout << "Maximum seaters available: 6 seaters" << endl;
        cnt ++;
        if(cnt == 2)
        {
            cout << "\n" << endl;
            cout << "Thank you for visiting Dining Table Store";
            return 0;
        }
    }

    if(roomLen >= (0.9 + space) && roomWid >= (0.9 + space))
    {
        cout << "-------------------------------------------------------------------" << endl;
        cout << "Recommended table: Square Lea" << endl;
        cout << "Maximum seaters available: 4 seaters" << endl;
        cnt ++;
        if(cnt == 2)
        {
            cout << "\n" << endl;
            cout << "Thank you for visiting Dining Table Store";
            return 0;
        }
    }
```

```cpp

    if(roomLen >= (0.76 + space) && roomWid >= (0.76 + space))
    {
        cout << "-------------------------------------------------------------------" << endl;
        cout << "Recommended table: Square Finn" << endl;
        cout << "Maximum seaters available: 2 seaters" << endl;
        cnt ++;
        if(cnt == 2)
        {
            cout << "\n" << endl;
            cout << "Thank you for visiting Dining Table Store";
            return 0;
        }
    }

    if(cnt == 0)
    {
        cout << "-------------------------------------------------------------------" << endl;
        cout << "Dear customers, sorry, none of the tables fulfil your requirement." << endl;
        cout << "\n" << endl;
        cout << "Thank you for visiting Dining Table Store";
    }

    return 0;
}
```

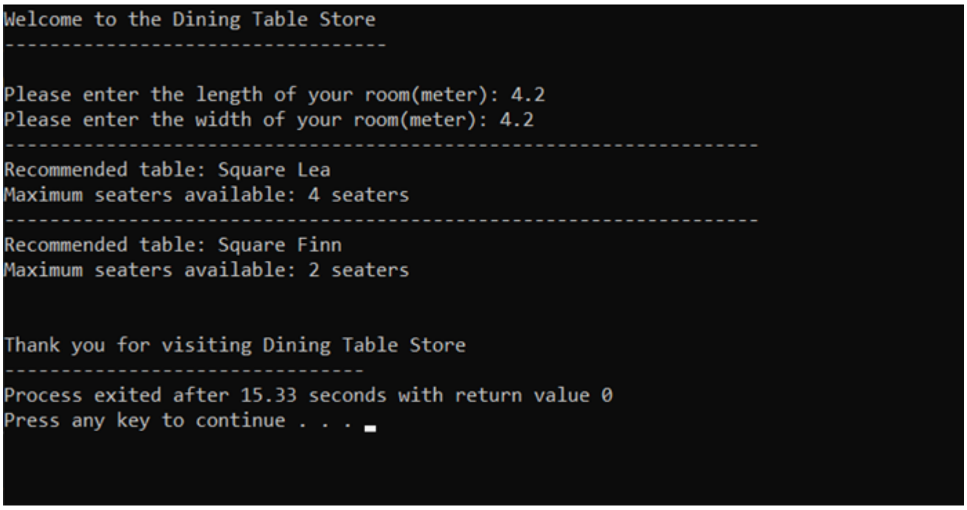
6.0) Sample Output



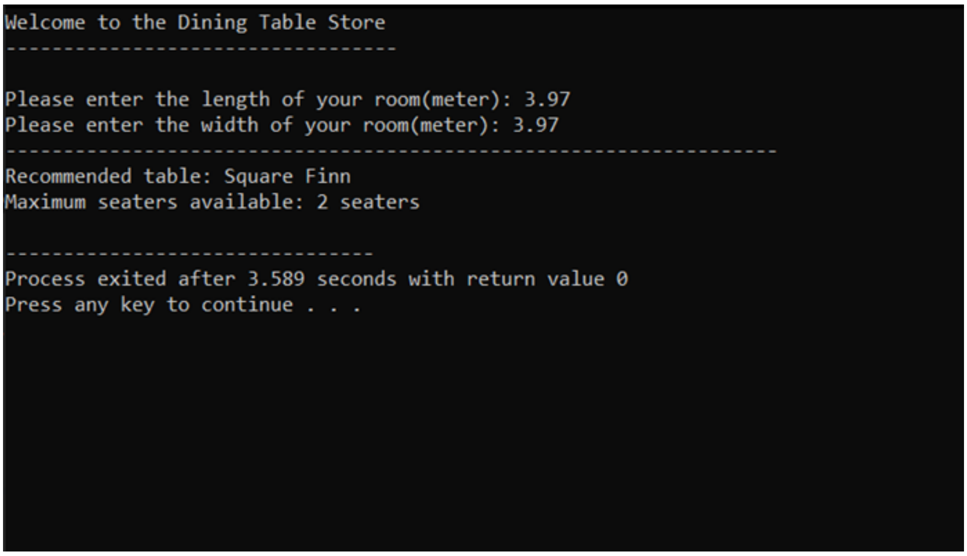
```
Welcome to the Dining Table Store
---------------------------------

Please enter the length of your room(meter): 6.3
Please enter the width of your room(meter): 4.3
-------------------------------------------------------------------
Recommended table: Rectangle Shade
Maximum seaters available: 12 seaters
-------------------------------------------------------------------
Recommended table: Rectangle Bertha
Maximum seaters available: 8 seaters


Thank you for visiting Dining Table Store
--------------------------------
Process exited after 17.85 seconds with return value 0
Press any key to continue . . .
```



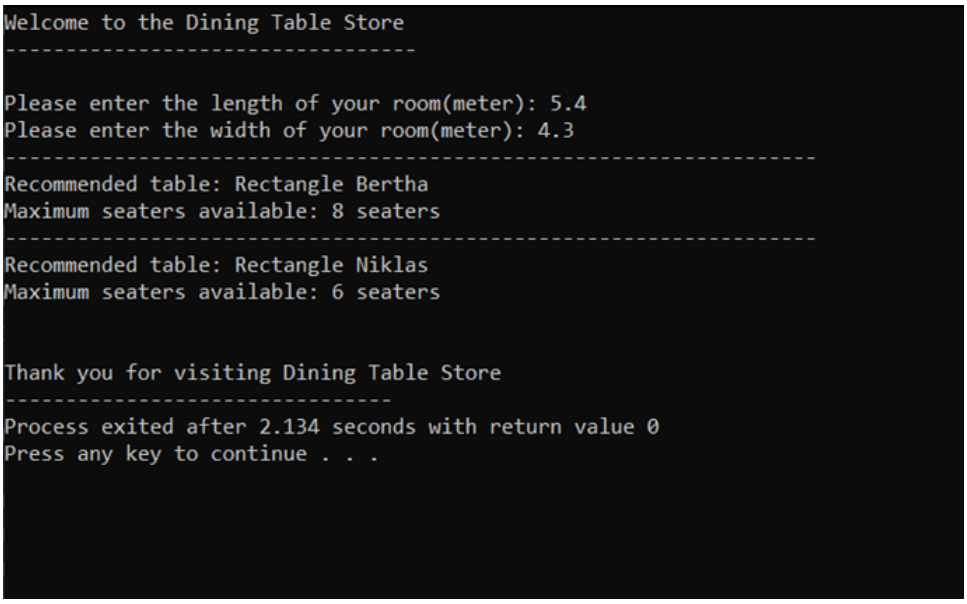
```
Welcome to the Dining Table Store
---------------------------------

Please enter the length of your room(meter): 6.2
Please enter the width of your room(meter): 4.6
-------------------------------------------------------------------
Recommended table: Rectangle Shade
Maximum seaters available: 12 seaters
-------------------------------------------------------------------
Recommended table: Oval Stefan
Maximum seaters available: 10 seaters


Thank you for visiting Dining Table Store
--------------------------------
Process exited after 15.31 seconds with return value 0
Press any key to continue . . .
```



```
Welcome to the Dining Table Store
----------------------------------

Please enter the length of your room(meter): 4.2
Please enter the width of your room(meter): 4.2
-------------------------------------------------------------------
Recommended table: Square Lea
Maximum seaters available: 4 seaters
-------------------------------------------------------------------
Recommended table: Square Finn
Maximum seaters available: 2 seaters


Thank you for visiting Dining Table Store
--------------------------------
Process exited after 15.33 seconds with return value 0
Press any key to continue . . .
```



```
Welcome to the Dining Table Store
----------------------------------

Please enter the length of your room(meter): 3.97
Please enter the width of your room(meter): 3.97
-------------------------------------------------------------------
Recommended table: Square Finn
Maximum seaters available: 2 seaters

--------------------------------
Process exited after 3.589 seconds with return value 0
Press any key to continue . . .
```



```
Welcome to the Dining Table Store
----------------------------------

Please enter the length of your room(meter): 5.4
Please enter the width of your room(meter): 4.3
-------------------------------------------------------------------
Recommended table: Rectangle Bertha
Maximum seaters available: 8 seaters
-------------------------------------------------------------------
Recommended table: Rectangle Niklas
Maximum seaters available: 6 seaters


Thank you for visiting Dining Table Store
--------------------------------

Process exited after 2.134 seconds with return value 0
Press any key to continue . . .
```