```cpp
// ========================================================================
// ME5015 StuID:NNNNNN      NAME::_______________
// Software written by Boguslaw Cyganek (C) to be used with the book:
// INTRODUCTION TO PROGRAMMING WITH C++ FOR ENGINEERS
// ========================================================================
// Revised ME5015-HW1ref:: for testing replit.com  (Date:2022-Sep-23)
// listing 3.2 in book (page. 53~ 56)
//
#include <vector>→→→// A header to use std::vector
#include <iostream>→→→// Headers for input and output
#include <iomanip>→→→// and output formatting
#include <cmath>→→→// For math functions

// Introduce these, to write vector instead of std::vector
using std::cout, std::cin, std::endl, std::vector;

int main()
{

  cout << "(ME5015-CPPbook::R0)Enter your GPA grades:(1.0 ~5.0 max):" << endl;
  cout << "\n\tNote: If GPA grade < 2.0, DOWN. Not be counted !!" << endl;
  vector< double > studentGradeVec;→// An empty vector of doubles
  int stuCount =0;
→// Collect student's grades
→for( ;; )
→{
→→double grade {};

→→cin >> grade;

→→// If ok, push new grade at the end of the vector
→→if( grade >= 2.0 && grade <= 5.0 ){
    stuCount++;
→→→studentGradeVec.push_back( grade );
    }
→
→→cout << "OK count="<< stuCount <<  "\tEnter more? [y/n] ";
→→char ans {};
→→cin >> ans;

→→if( ans == 'n' || ans == 'N' )
→→→break;→→// the way to exit the loop
→}

→// Ok, if there are any grades compute the average
→if( studentGradeVec.size() > 0 )
→{
→→double sum { 0.0 };
→→// Add all the grades
→→for( auto g : studentGradeVec )
→→→sum += g;

→→double av = sum / studentGradeVec.size();

→→double finalGrade {};

→→// Let it adjust
→→if( av < 3.0 )
→→{
→→→finalGrade = 2.0;
→→}
→→else
→→{
→→→double near_int = std::floor( av );→// get integer part
→→→double frac→→= av - near_int;→→// get only the fraction

→→→double adjust { 0.5 };→// new adjustment value

→→→if( frac < 0.25 )
→→→→adjust = 0.0;
→→→else if( frac > 0.75 )
→→→→adjust = 1.0;

```

```
74      ->->->finalGrade = near_int + adjust;
75      ->->}
76      ->->cout << "\n\tNote: Lucky formula::: finalGrade = near_int + adjust\n ";
77      ->->cout->>"Final grade: "
78      ->->->// >> std::fixed << std::setw( 3 ) << std::setprecision( 1 ) //ORG.mode
79      ->>std::fixed << std::setw( 6 ) << std::setprecision( 2 )
80      ->->->->>finalGrade << " Students Count="<< stuCount <<endl;
81      ->}
82
83      ->return 0;
84      }
85      /* --output
86      (ME5015-CPPbook::R0)Enter your GPA grades:(1.0 ~5.0 max):
87
88          Note: If GPA grade < 2.0, DOWN. Not be counted !!
89      4.67
90      OK count=1  Enter more? [y/n] y
91      5.999
92      OK count=1  Enter more? [y/n] y
93      4.95
94      OK count=2  Enter more? [y/n] y
95      1.5
96      OK count=2  Enter more? [y/n] y
97      3.99
98      OK count=3  Enter more? [y/n] y
99      4.6789
100     OK count=4  Enter more? [y/n] y
101     4.9
102     OK count=5  Enter more? [y/n] n
103
104         Note: Lucky formula::: finalGrade = near_int + adjust
105      Final grade:    4.50 Students Count=5
106     */
107
```