A Preparation Guide for Admission Examination/Technical Interview

Master of Science in Computing and Data Analytics (MSc CDA)

Faculty of Science

Saint Mary's University

Table of Contents

How to use this guide

This guide will help students who are applying to the MSc program in Computing Science and Data Analytics (MSc CDA) prepare for the admission examination/technical interview that will be conducted to assess prospective students' skills and qualifications to enter the program.

It consists of two parts:

Part 1 of this guide shows practice questions and their sample solutions with links to the related YouTube videos. Students should carefully read these questions and attempt to solve them all on their own. They should then compare their answers with the ones that are provided. We also recommend that the students type these answers in an IDE of their choice, compile, run, and test the programs to make sure the output will exactly match the sample output.

Part 2 of this guide is a list of questions that we have used in the past and are similar to those that will be asked in the technical interview. While we do NOT provide solutions to the Part 2 questions, students are encouraged to study all the questions so that the output will exactly match the sample output.

Part 2 contains 3 sets of questions, grouped by type of questions.

- Set 1 is a group of questions with simple/nested loops and scientific/mathematical formulae.
- Set 2 contains questions that require certain data structures (e.g., arrays, lists, classes).
- Set 3 consists of database questions that ask for queries to summarize and/or extract information from given data.

While not mandatory, students are STRONGLY encouraged to solve and bring solutions to at least one question from each set provided in Part 2 to the interview. The interviewer will review the code and give you a chance to highlight your programming skills.

Admission Examination/Technical Interview

- When your application is complete with all the required documents, we will review your entire application package.
- If it satisfies all the minimum requirements, then you will be invited to schedule a timeslot for the admission examination/technical interview.
- You may write programs in any language. However, we strongly recommend Java, C++, C, Python or C#.
- As mentioned above, you are encouraged to bring solutions to the questions listed in this guide, and you may be asked questions about your solutions.
- During the interview, you may also use this guide and solutions to the questions listed in it as well as any materials that you have prepared.
- You will not, however, be allowed to use any coding assistance such as AI tools, online forums, or any interactive help from others (e.g., text message).
- We will often ask applicants to share their screens during the interview and keep the camera on all the time.

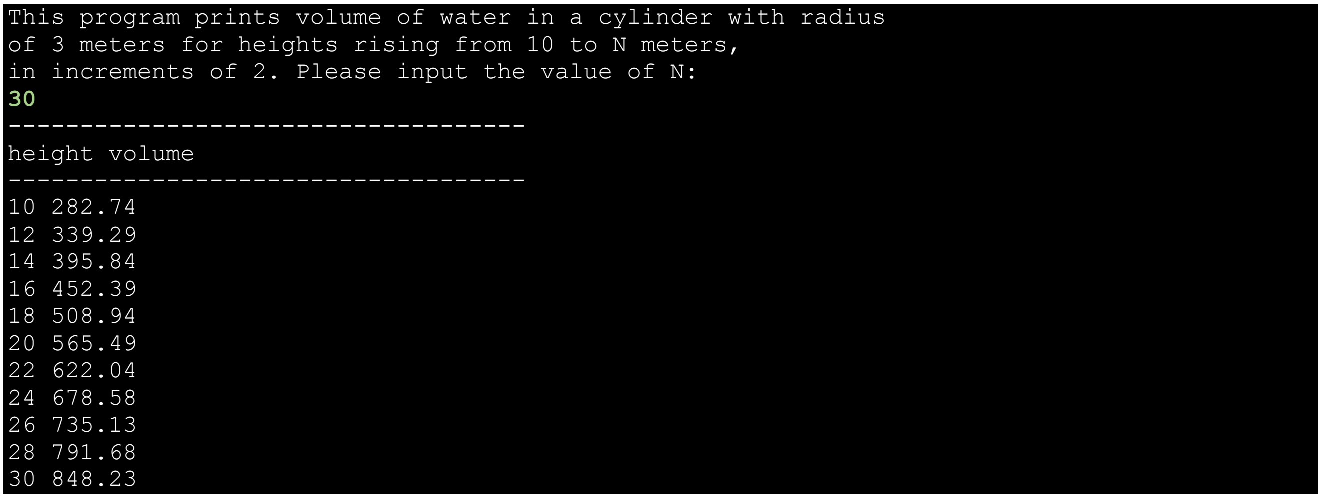
Part 1: Practice Questions (PQs)

PQ 1

Write a program that prints a table of how much volume of water a cylinder (with a radius r of 3 meters) is holding, as the height of water rises from 10 to N meters in increments of 2, where the value of N is input by the user (use the formula $volume = \pi \times r^2$)

The following is a sample session (Do not worry about rounding off the values):

Sample Run



```
This program prints volume of water in a cylinder with radius
of 3 meters for heights rising from 10 to N meters,
in increments of 2. Please input the value of N:
30
------------------------------------
height volume
------------------------------------
10 282.74
12 339.29
14 395.84
16 452.39
18 508.94
20 565.49
22 622.04
24 678.58
26 735.13
28 791.68
30 848.23
```

Solution (Video with a solution: http://youtu.be/IfWUM9VGndY):

```
import java.util.Scanner;
public class Q4
{
    public static void main(String [] a)
    {
        Scanner k = new Scanner(System.in);
        System.out.print("This program prints volume of water " +
        "in a cylinder with radius\nof 3 meters for heights ranging from " +
        "10 to N meters, in increments of 2.\nPlease input the value of N: ");
        int N = k .nextInt();
        System.out.println("----------------------");
        System.out.println("height\t\tvolume");
        System.out.println("----------------------");
        for(int i = 10; i <= N; i +=2)
        {
            double volume = i*Math.PI*Math.pow(3,2);
            System.out.printf("%d\t\t%.2f\n",i,volume);
        }
    }
}
```

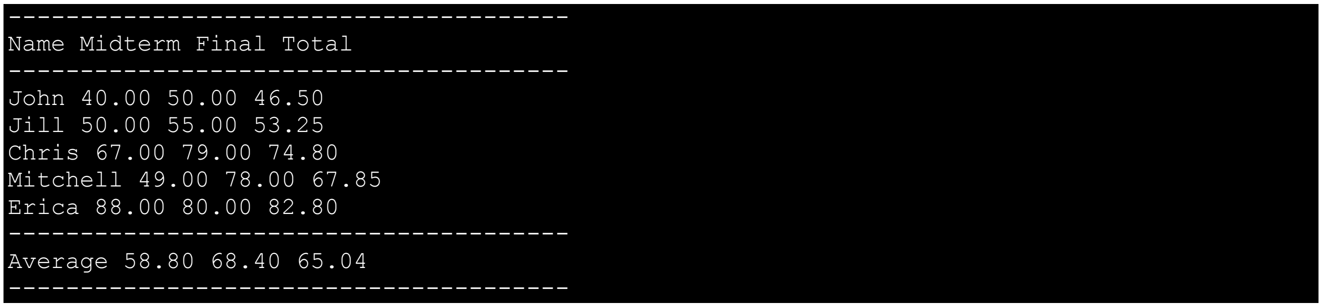
PQ 2

Write a program that:

- reads a number N
- uses one array of Strings of length N to store student names, an array of doubles of length N to store the marks in the midterm (marks range from 0-100 and weighted 35% of total), and an array of doubles to store the marks in final examination (marks range from 0-100 and weighted 65% of total)
- reads a list of student names, midterm and final examination marks, stores them in the corresponding arrays
- calculates the total marks for each student
- keeps track of the sum of marks for the midterm and final examination
- prints all student names, midterm, final examination, and total marks, as well as the grade in a table
- At the end, prints the average for midterm, final examination, and total marks, as well as the average grade

(Do not worry about having exactly two decimal points and other fancy number formatting)

Sample Output



```
---------------------------------------
Name Midterm Final Total
---------------------------------------
John 40.00 50.00 46.50
Jill 50.00 55.00 53.25
Chris 67.00 79.00 74.80
Mitchell 49.00 78.00 67.85
Erica 88.00 80.00 82.80
---------------------------------------
Average 58.80 68.40 65.04
---------------------------------------
```

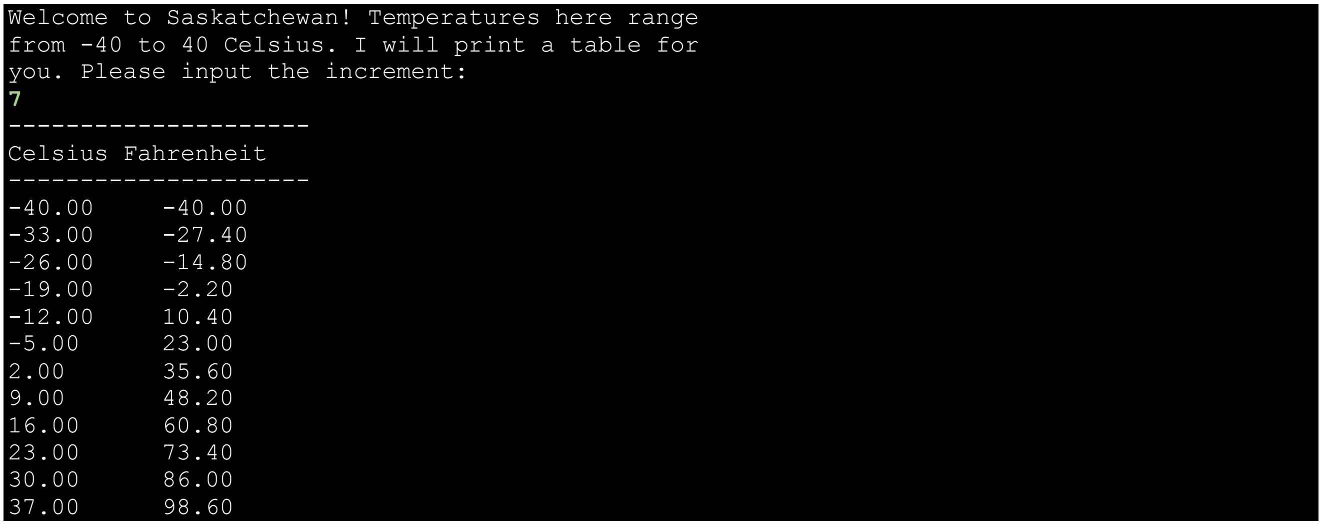
Solution:

```
import java.util.Scanner;
public class QF5
{
    public static void main(String [] a)
    {
        Scanner k = new Scanner(System.in);

        System.out.print("How many students? ");
        int n = k.nextInt();

        String [] names = new String[n];
        double [] marks1 = new double[n];
        double [] marks2 = new double[n];

        readArray(k,names,marks1,marks2);
        writeArray(names,marks1,marks2);
        writeAverage(marks1,marks2);
    }

    public static void writeAverage(double [] marks1, double [] marks2)
    {
        double sum1 = 0, sum2 = 0;
        for(int i = 0; i < marks1.length; i++)
        {
            sum1 = sum1 + marks1[i];
            sum2 = sum2 + marks2[i];
        }

        System.out.println("\tAvg\t\t"+(sum1/marks1.length)
           + "\t"+(sum2/marks2.length)
           + "\t"+((sum1*0.35+sum2*0.65)/marks1.length));
    }

    public static void readArray(Scanner k, String [] names,
           double [] marks1, double [] marks2)
    {
        for(int i = 0; i < names.length; i++)
        {
            System.out.print("Name of student " + i + ":");
            names[i] = k.next();
            System.out.print("Midterm marks for " + names[i] + ":");
            marks1[i] = k.nextDouble();
            System.out.print("Final marks for " + names[i] + ":");
            marks2[i] = k.nextDouble();
        }
    }

    public static void writeArray(String [] names,
           double [] marks1, double [] marks2)
    {
        System.out.println("---------------------------------------");
        System.out.println("No.\tName\t\tMid\tFin\tTot");
        System.out.println("---------------------------------------");

        for(int i = 0; i < names.length; i++)
        {
           double tot = marks1[i]*0.35+marks2[i]*0.65;
            System.out.println(i+"\t"+names[i] + "\t\t" + marks1[i]
                                            +"\t"+marks2[i]+"\t"+tot);
        }
        System.out.println("---------------------------------------");
    }
}
```

PQ 3

Write a program that prints a table of conversion from Celsius (C) to Fahrenheit (F) for American visitors to Saskatchewan for temperatures ranging from -40 to +40 in increments of “inc”, where inc will be input by the user. $F = \frac{9}{5} \times C + 32$. A sample session is given below. Do not worry about the number of decimal places.

Sample Run



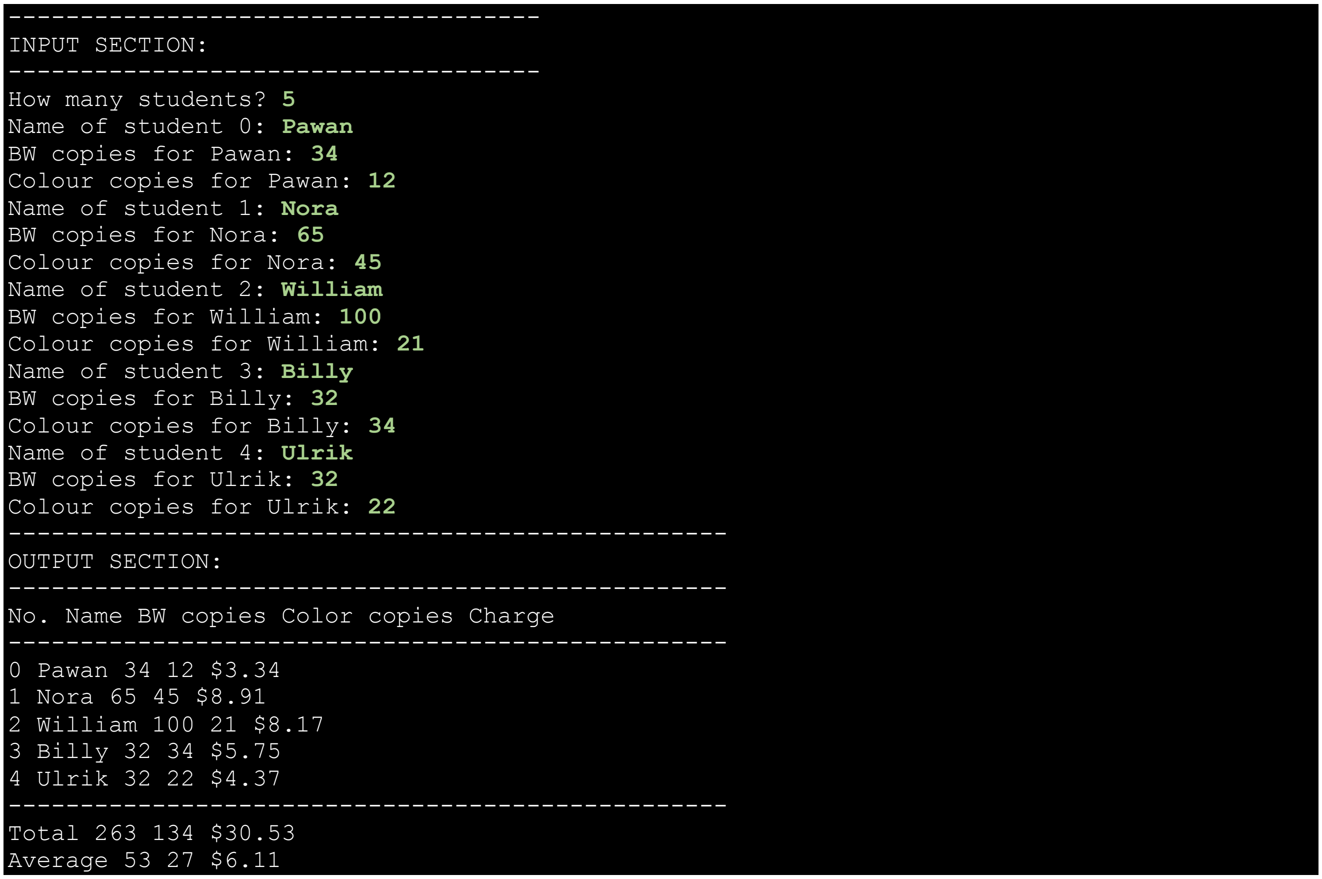
```
Welcome to Saskatchewan! Temperatures here range
from -40 to 40 Celsius. I will print a table for
you. Please input the increment:
7
---------------------
Celsius Fahrenheit
---------------------
-40.00     -40.00
-33.00     -27.40
-26.00     -14.80
-19.00     -2.20
-12.00     10.40
-5.00      23.00
2.00       35.60
9.00       48.20
16.00      60.80
23.00      73.40
30.00      86.00
37.00      98.60
```

PQ 4

Write a program that:

- uses one array of Strings of length N to store student names, an array of integers of length N to store the number of black and white (BW) papers printed (charged at $0.05 per page), and an array of integers to store the number of colour papers printed (charged at $0.10 per page),
- reads a list of student names, numbers of BW and colour pages printed, stores them in the corresponding arrays
- calculates the printing charge for each student which should also include 15% HST
- keeps track of the sum of BW and colour pages printed
- prints all student names, the numbers of BW and colour pages printed, as well as the printing charge
- at the end, prints:
- total numbers of BW and colour pages printed, as well as the printing charge
- average numbers of BW and colour pages printed, as well as the printing charge
- (Do not worry about having exactly two decimal points and other fancy number formatting)

Sample Run



```
-------------------------------------
INPUT SECTION:
-------------------------------------
How many students? 5
Name of student 0: Pawan
BW copies for Pawan: 34
Colour copies for Pawan: 12
Name of student 1: Nora
BW copies for Nora: 65
Colour copies for Nora: 45
Name of student 2: William
BW copies for William: 100
Colour copies for William: 21
Name of student 3: Billy
BW copies for Billy: 32
Colour copies for Billy: 34
Name of student 4: Ulrik
BW copies for Ulrik: 32
Colour copies for Ulrik: 22
--------------------------------------------------
OUTPUT SECTION:
--------------------------------------------------
No. Name BW copies Color copies Charge
--------------------------------------------------
0 Pawan 34 12 $3.34
1 Nora 65 45 $8.91
2 William 100 21 $8.17
3 Billy 32 34 $5.75
4 Ulrik 32 22 $4.37
--------------------------------------------------
Total 263 134 $30.53
Average 53 27 $6.11
```

There is no video that develops the solutions for PQ3 or PQ4, but the following set of three videos and corresponding code will help you get there.

Video I: http://youtu.be/wHnjXqRe7Ic

Program developed in the video:

```
import java.util.Scanner;
public class StudentNameArray
{
    public static void main(String [] a)
    {
        Scanner k = new Scanner(System.in);

        System.out.print("How many students? ");
        int n = k.nextInt();
        String [] names = new String[n];

        for(int i = 0; i < names.length; i++)
        {
            System.out.print("Name of student " + i + ":");
            names[i] = k.next();
        }

        for(int i = 0; i < names.length; i++)
        {
            System.out.println("The name of student " + i + " is " + names[i]);
        }
    }
}
```

Video II: http://youtu.be/D_j3kT8BSMU

Program developed in the video:

```
import java.util.Scanner;
public class StudentNameArrayFun
{
    public static void main(String [] a)
    {
        Scanner k = new Scanner(System.in);

        System.out.print("How many students? ");
        int n = k.nextInt();
        String [] names = new String[n];

        readArray(k,names);
        writeArray(names);
    }

    public static void readArray(Scanner k, String [] names)
    {
        for(int i = 0; i < names.length; i++)
        {
            System.out.print("Name of student " + i + ":");
            names[i] = k.next();
        }
    }


    public static void writeArray(String [] names)
    {
        for(int i = 0; i < names.length; i++)
        {
            System.out.println("The name of student " + i + " is " + names[i]);
        }
    }
}
```

Video III: *http://youtu.be/n8-opxX8YHg*

Program developed in the video:

```
import java.util.Scanner;
public class StudentMarksSummary
{
    public static void main(String [] a) {
        Scanner k = new Scanner(System.in);

        System.out.print("How many students? ");
        int n = k.nextInt();

        String [] names = new String[n];
        double [] marks = new double[n];

        readArray(k,names,marks);
        writeArray(names,marks);
        writeAverage(marks);
        writeMaximum(marks);
    }

    public static void writeAverage(double [] marks) {
        double sum = 0;
        for(int i = 0; i < marks.length; i++)
        {
            sum = sum + marks[i];
        }
        System.out.println("\tAvg\t\t"+(sum/marks.length));
    }

    public static void writeMaximum(double [] marks){
        double max = marks[0];
        for(int i = 1; i < marks.length; i++)
        {
            if(max < marks[i])
            {
                max = marks[i];
            }
        }
        System.out.println("\tMax\t\t"+max);
    }

    public static void readArray(Scanner k, String [] names, double [] marks){
        for(int i = 0; i < names.length; i++)
        {
            System.out.print("Name of student " + i + ":");
            names[i] = k.next();
            System.out.print("Marks for " + names[i] + ":");
            marks[i] = k.nextDouble();
        }
    }

    public static void writeArray(String [] names, double [] marks) {
        System.out.println("---------------------------------------");
        System.out.println("No.\tName\t\tMarks");
        System.out.println("---------------------------------------");

        for(int i = 0; i < names.length; i++)
        {
            System.out.println(i+"\t"+names[i] + "\t\t" + marks[i]);
        }
        System.out.println("---------------------------------------");
    }
}
```

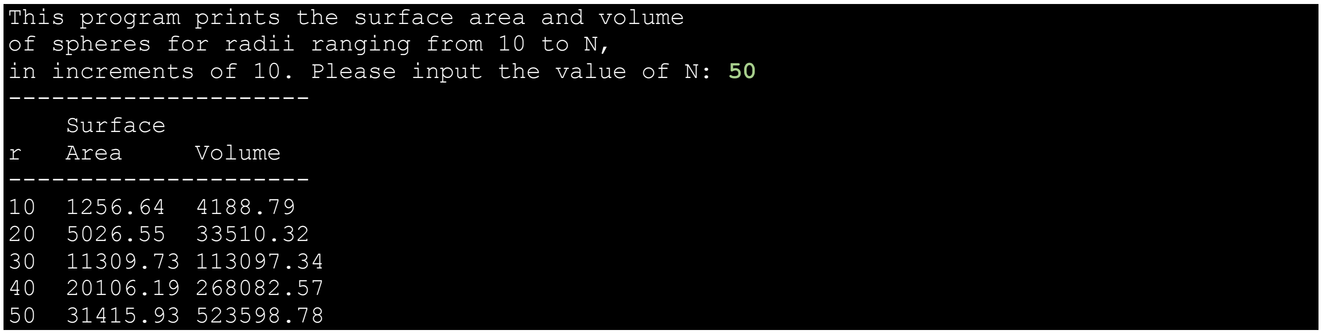
Part 2: Sample Questions (SQs) for Admissions Exam/Technical Interview

Set 1

SQ 1.1

Write a program that prints a table of surface area and volume of spheres for radii ranging from 10 to N in increments of 10, where the value of N is input by the user. Please find out the formulae for the surface area and volume of a sphere on your own.

Sample Run



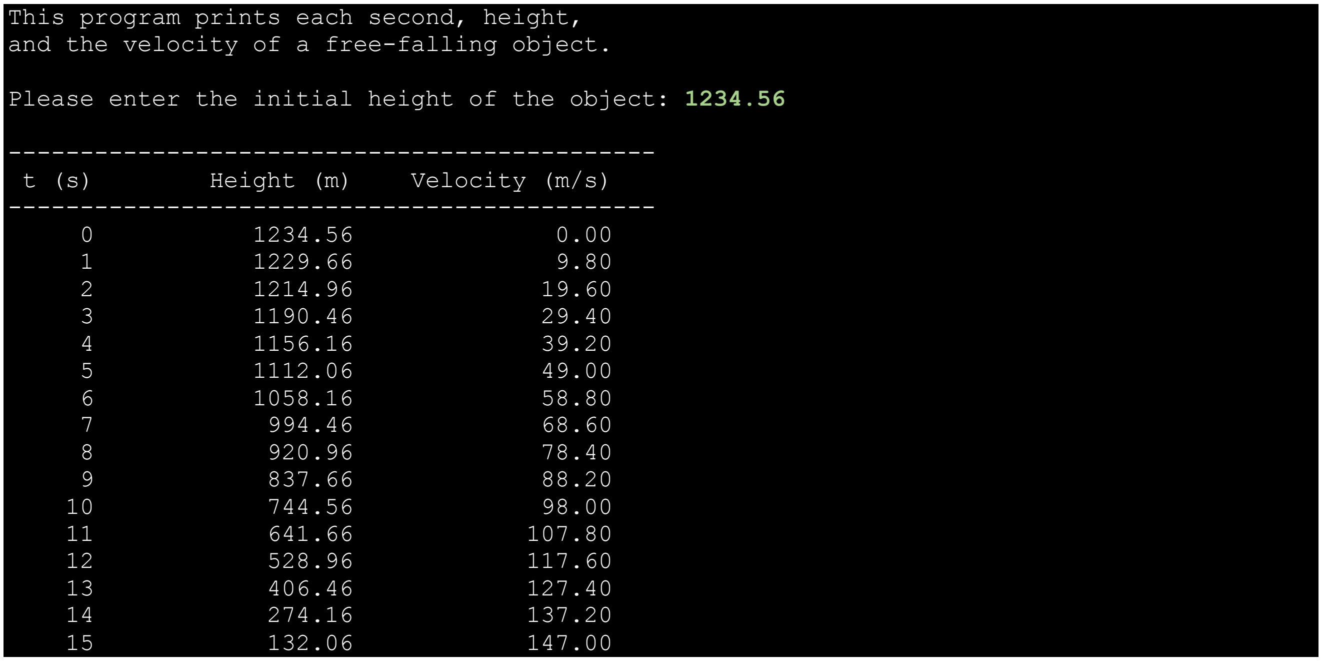
```
This program prints the surface area and volume
of spheres for radii ranging from 10 to N,
in increments of 10. Please input the value of N: 50
---------------------
    Surface
r   Area     Volume
---------------------
10  1256.64  4188.79
20  5026.55  33510.32
30  11309.73 113097.34
40  20106.19 268082.57
50  31415.93 523598.78
```

SQ 1.2

Write a program that prints a table that shows each second, the height from the ground (meters), and the velocity (m/s) of a free-falling object from an initial height (metres). The initial height is given by the user.

Please find out the necessary formulae and use 9.8 (m/s2) for the gravitational acceleration.

Sample Run



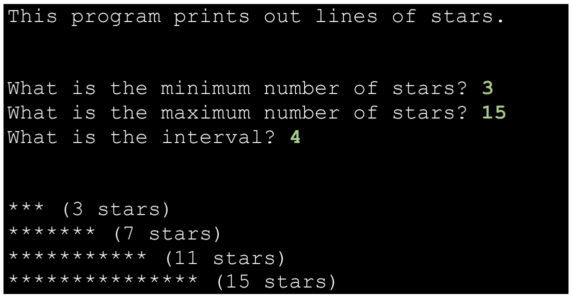
```
This program prints each second, height,
and the velocity of a free-falling object.

Please enter the initial height of the object: 1234.56

---------------------------------------------
 t (s)        Height (m)    Velocity (m/s)
---------------------------------------------
     0           1234.56              0.00
     1           1229.66              9.80
     2           1214.96             19.60
     3           1190.46             29.40
     4           1156.16             39.20
     5           1112.06             49.00
     6           1058.16             58.80
     7            994.46             68.60
     8            920.96             78.40
     9            837.66             88.20
    10            744.56             98.00
    11            641.66            107.80
    12            528.96            117.60
    13            406.46            127.40
    14            274.16            137.20
    15            132.06            147.00
```

SQ 1.3

Write a program that prints out lines of stars, based on the values determined by the user-input. The program first asks for the minimum number of stars, the maximum number of stars, and the interval. It then prints lines of stars, starting with a line with the minimum number of stars, and ending with a line with the maximum number of stars. The increment of the number of stars is specified by the interval value. At the end of each line, it should also print how many stars are printed.

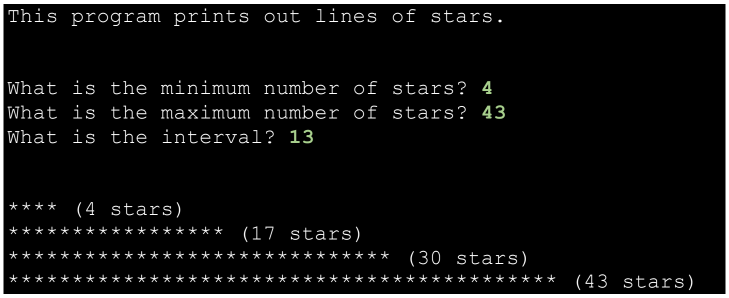
Sample Runs

Sample Run #1



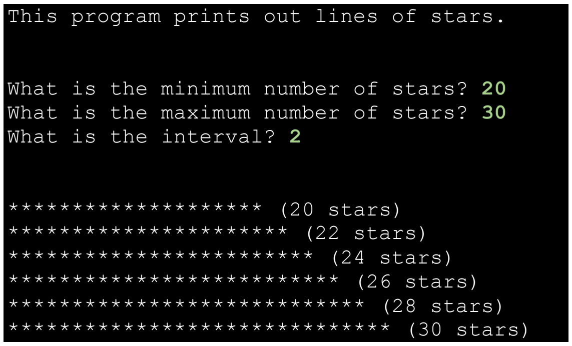
```
This program prints out lines of stars.


What is the minimum number of stars? 3
What is the maximum number of stars? 15
What is the interval? 4


*** (3 stars)
******* (7 stars)
*********** (11 stars)
*************** (15 stars)
```

Sample Run #2



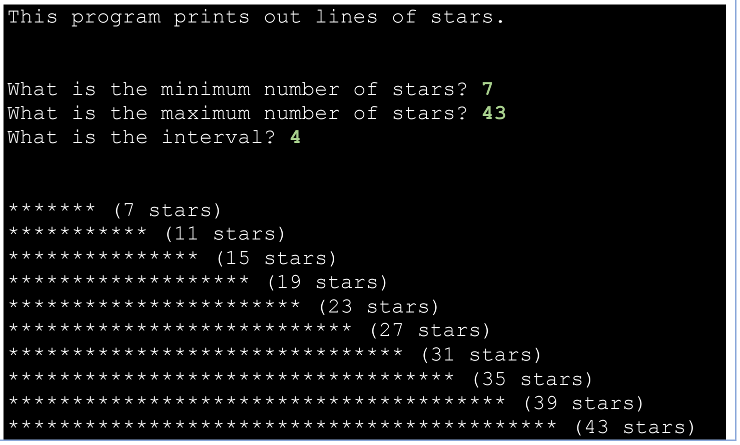
```
This program prints out lines of stars.


What is the minimum number of stars? 4
What is the maximum number of stars? 43
What is the interval? 13


**** (4 stars)
***************** (17 stars)
****************************** (30 stars)
******************************************* (43 stars)
```

Sample Run #3



```
This program prints out lines of stars.


What is the minimum number of stars? 20
What is the maximum number of stars? 30
What is the interval? 2


******************** (20 stars)
********************** (22 stars)
************************ (24 stars)
************************** (26 stars)
**************************** (28 stars)
****************************** (30 stars)
```

Sample Run #4



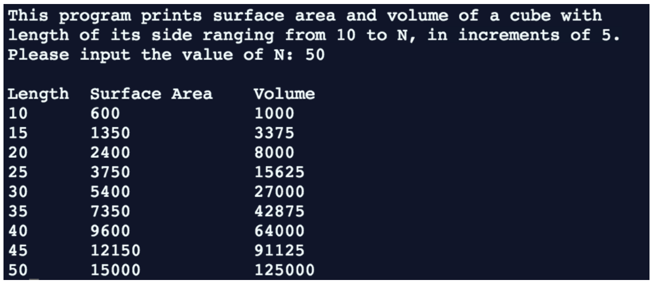
```
This program prints out lines of stars.


What is the minimum number of stars? 7
What is the maximum number of stars? 43
What is the interval? 4


******* (7 stars)
*********** (11 stars)
*************** (15 stars)
******************* (19 stars)
*********************** (23 stars)
*************************** (27 stars)
******************************* (31 stars)
*********************************** (35 stars)
*************************************** (39 stars)
******************************************* (43 stars)
```

SQ 1.4

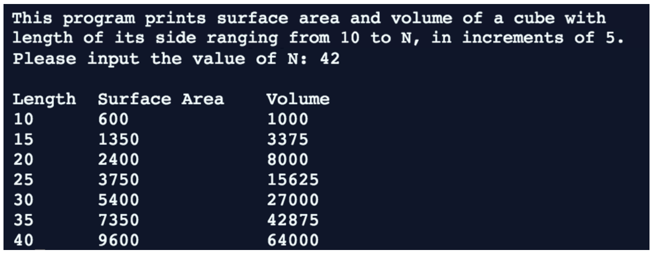
Write a program to calculate the volume and surface area of a cube having equal faces with equal length, breadth and height, accepting the length of the side ranging from 10 to N in increments of 5, where the value of N is input by the user.

Please find out the formulae for the volume and surface area of a cube.

Sample Runs



```
This program prints surface area and volume of a cube with
length of its side ranging from 10 to N, in increments of 5.
Please input the value of N: 50

Length  Surface Area    Volume
10      600             1000
15      1350            3375
20      2400            8000
25      3750            15625
30      5400            27000
35      7350            42875
40      9600            64000
45      12150           91125
50      15000           125000
```



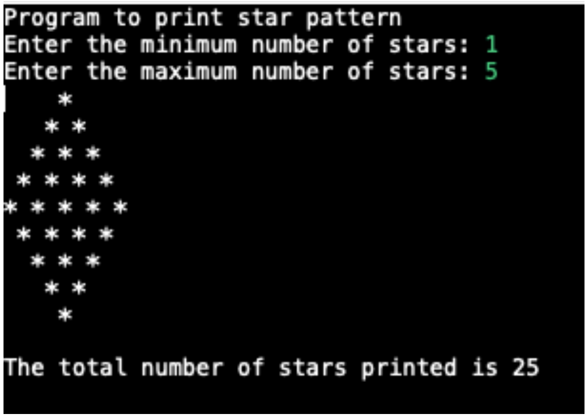
```
This program prints surface area and volume of a cube with
length of its side ranging from 10 to N, in increments of 5.
Please input the value of N: 42

Length  Surface Area    Volume
10      600             1000
15      1350            3375
20      2400            8000
25      3750            15625
30      5400            27000
35      7350            42875
40      9600            64000
```

SQ 1.5

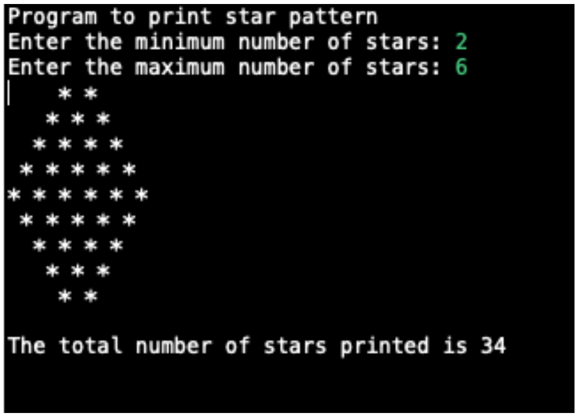
Write a program that prints out lines of stars, based on the values determined by the user-input.

The program should print the number of stars that increases on each line from the minimum number until it reaches the maximum number and then decreases until it goes back to the minimum number. After printing out the lines of stars, it should also print the total number of stars printed.

Sample Runs



```
Program to print star pattern
Enter the minimum number of stars: 1
Enter the maximum number of stars: 5
    *
   * *
  * * *
 * * * *
* * * * *
 * * * *
  * * *
   * *
    *

The total number of stars printed is 25
```

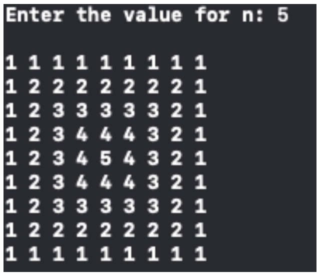


```
Program to print star pattern
Enter the minimum number of stars: 2
Enter the maximum number of stars: 6
    * *
   * * *
  * * * *
 * * * * *
* * * * * *
 * * * * *
  * * * *
   * * *
    * *

The total number of stars printed is 34
```

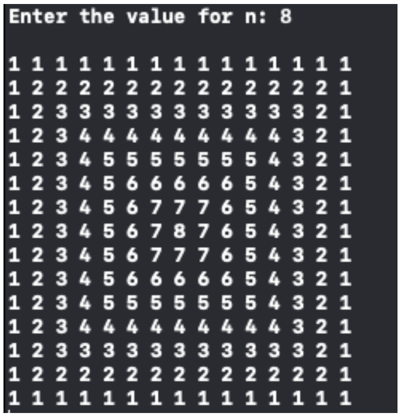
SQ 1.6

Write a program that prints the following pattern for a given input *n* containing numbers from 1 to *n*.

Sample Runs



```
Enter the value for n: 5

1 1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 2 1
1 2 3 3 3 3 3 2 1
1 2 3 4 4 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 4 4 3 2 1
1 2 3 3 3 3 3 2 1
1 2 2 2 2 2 2 2 1
1 1 1 1 1 1 1 1 1
```

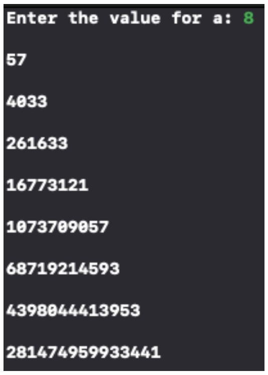


```
Enter the value for n: 8

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 2 2 2 2 2 2 2 1
1 2 3 3 3 3 3 3 3 3 3 3 3 2 1
1 2 3 4 4 4 4 4 4 4 4 4 3 2 1
1 2 3 4 5 5 5 5 5 5 5 4 3 2 1
1 2 3 4 5 6 6 6 6 6 5 4 3 2 1
1 2 3 4 5 6 7 7 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 7 7 6 5 4 3 2 1
1 2 3 4 5 6 6 6 6 6 5 4 3 2 1
1 2 3 4 5 5 5 5 5 5 5 4 3 2 1
1 2 3 4 4 4 4 4 4 4 4 4 3 2 1
1 2 3 3 3 3 3 3 3 3 3 3 3 2 1
1 2 2 2 2 2 2 2 2 2 2 2 2 2 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

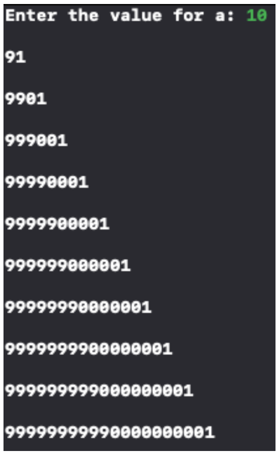
SQ 1.7

Write a program that prints the values for the formula $a^{2x} - a^{x} + 1$ for a given input *a* and *x* ranging from `1` to `a` by an increment of `1`.

Sample Runs



```
Enter the value for a: 8

57

4033

261633

16773121

1073709057

68719214593

4398044413953

281474959933441
```



```
Enter the value for a: 10

91

9901

999001

99990001

9999900001

999999000001

99999990000001

9999999900000001

999999999000000001

99999999990000000001
```

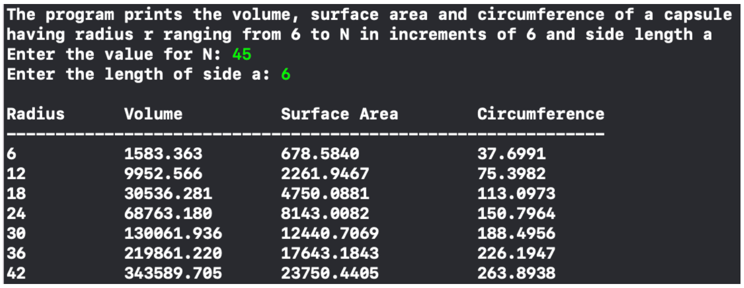
SQ 1.8

Write a program to calculate the volume, surface area, and circumference of a capsule having radius `r` ranging from 6 to `N` in increments of 6 and side length `a`, where `N` and `a` are determined by the user.

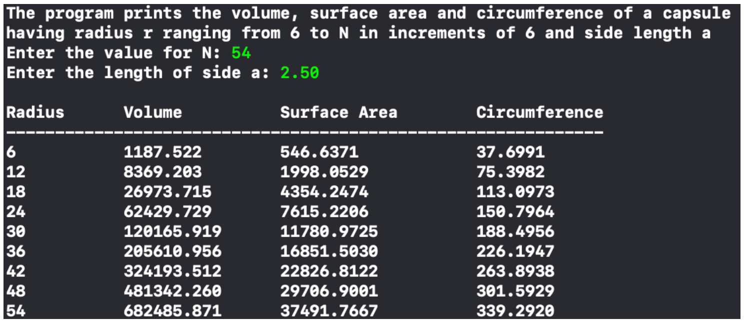
Please find out the formulae to calculate the volume, surface area and circumference of a capsule.

Note that the resulting values in the examples were based on the `PI` value of `3.14159265358979` and may differ accordingly if you use other values/built-in constants.

Sample Runs



```
The program prints the volume, surface area and circumference of a capsule
having radius r ranging from 6 to N in increments of 6 and side length a
Enter the value for N: 45
Enter the length of side a: 6

Radius      Volume          Surface Area        Circumference
-------------------------------------------------------------
6           1583.363        678.5840            37.6991
12          9952.566        2261.9467           75.3982
18          30536.281       4750.0881           113.0973
24          68763.180       8143.0082           150.7964
30          130061.936      12440.7069          188.4956
36          219861.220      17643.1843          226.1947
42          343589.705      23750.4405          263.8938
```



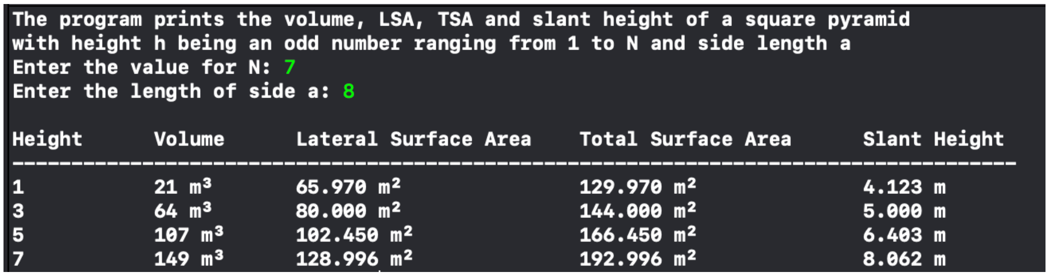
```
The program prints the volume, surface area and circumference of a capsule
having radius r ranging from 6 to N in increments of 6 and side length a
Enter the value for N: 54
Enter the length of side a: 2.50

Radius      Volume          Surface Area        Circumference
-------------------------------------------------------------
6           1187.522        546.6371            37.6991
12          8369.203        1998.0529           75.3982
18          26973.715       4354.2474           113.0973
24          62429.729       7615.2206           150.7964
30          120165.919      11780.9725          188.4956
36          205610.956      16851.5030          226.1947
42          324193.512      22826.8122          263.8938
48          481342.260      29706.9001          301.5929
54          682485.871      37491.7667          339.2920
```

SQ 1.9

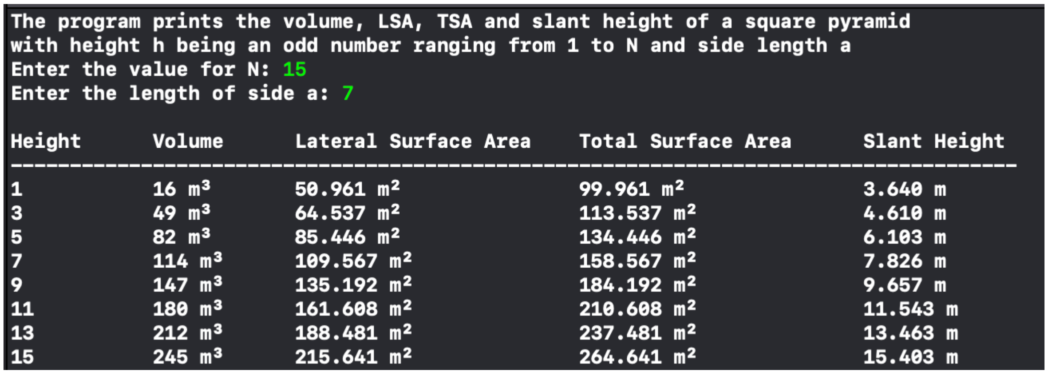
Write a program to calculate the volume, Lateral Surface Area (LSA), Total Surface Area (TSA), and Slant Height (SH) of a square pyramid having height `h` being an odd number ranging from 1 to N and side length `a`, where `N` and `a` are inputs provided by the user.

Please find out the formulae for volume, LSA, TSA, and SH of a square pyramid.
(* Hint: Round off the value of Volume of the square pyramid)

Sample Runs



```
The program prints the volume, LSA, TSA and slant height of a square pyramid
with height h being an odd number ranging from 1 to N and side length a
Enter the value for N: 7
Enter the length of side a: 8

Height      Volume      Lateral Surface Area      Total Surface Area      Slant Height
--------------------------------------------------------------------------------------
1           21 m³       65.970 m²                 129.970 m²              4.123 m
3           64 m³       80.000 m²                 144.000 m²              5.000 m
5           107 m³      102.450 m²                166.450 m²              6.403 m
7           149 m³      128.996 m²                192.996 m²              8.062 m
```



```
The program prints the volume, LSA, TSA and slant height of a square pyramid
with height h being an odd number ranging from 1 to N and side length a
Enter the value for N: 15
Enter the length of side a: 7

Height      Volume      Lateral Surface Area      Total Surface Area      Slant Height
--------------------------------------------------------------------------------------
1           16 m³       50.961 m²                 99.961 m²               3.640 m
3           49 m³       64.537 m²                 113.537 m²              4.610 m
5           82 m³       85.446 m²                 134.446 m²              6.103 m
7           114 m³      109.567 m²                158.567 m²              7.826 m
9           147 m³      135.192 m²                184.192 m²              9.657 m
11          180 m³      161.608 m²                210.608 m²              11.543 m
13          212 m³      188.481 m²                237.481 m²              13.463 m
15          245 m³      215.641 m²                264.641 m²              15.403 m
```

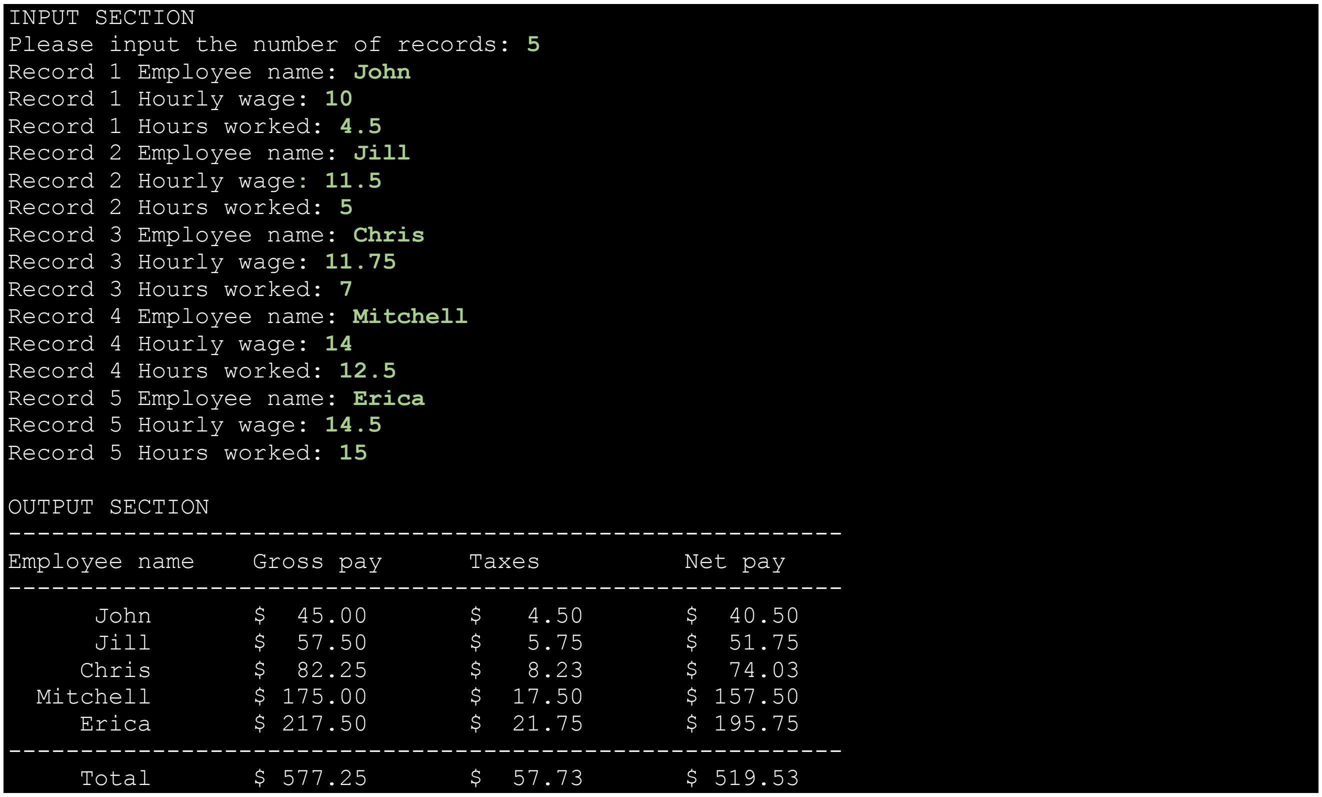
Set 2

SQ 2.1

Write a program that:

- asks the user for the number of records, N
- uses one array of Strings of length N to store employee names, an array of doubles of length N to store the hourly wage, and an array of doubles to store the number of hours worked.
- reads a list of N employee names, hourly wages, and hours worked, stores them in the corresponding arrays
- calculates the gross wages for each employee as the product of hourly wages and hours worked
- calculates the tax for each employee as 10% of the gross wages
- calculates net wages as gross wages minus tax
- keeps track of total gross wages, total taxes, and total net wages
- prints all N employee names, gross wages, tax, and net wages in a table
- At the end, prints total gross wages, total taxes, and total net wages

Sample Run



```
INPUT SECTION
Please input the number of records: 5
Record 1 Employee name: John
Record 1 Hourly wage: 10
Record 1 Hours worked: 4.5
Record 2 Employee name: Jill
Record 2 Hourly wage: 11.5
Record 2 Hours worked: 5
Record 3 Employee name: Chris
Record 3 Hourly wage: 11.75
Record 3 Hours worked: 7
Record 4 Employee name: Mitchell
Record 4 Hourly wage: 14
Record 4 Hours worked: 12.5
Record 5 Employee name: Erica
Record 5 Hourly wage: 14.5
Record 5 Hours worked: 15

OUTPUT SECTION
----------------------------------------------------------
Employee name    Gross pay      Taxes          Net pay
----------------------------------------------------------
      John       $  45.00       $   4.50       $  40.50
      Jill       $  57.50       $   5.75       $  51.75
     Chris       $  82.25       $   8.23       $  74.03
  Mitchell       $ 175.00       $  17.50       $ 157.50
     Erica       $ 217.50       $  21.75       $ 195.75
----------------------------------------------------------
     Total       $ 577.25       $  57.73       $ 519.53
```

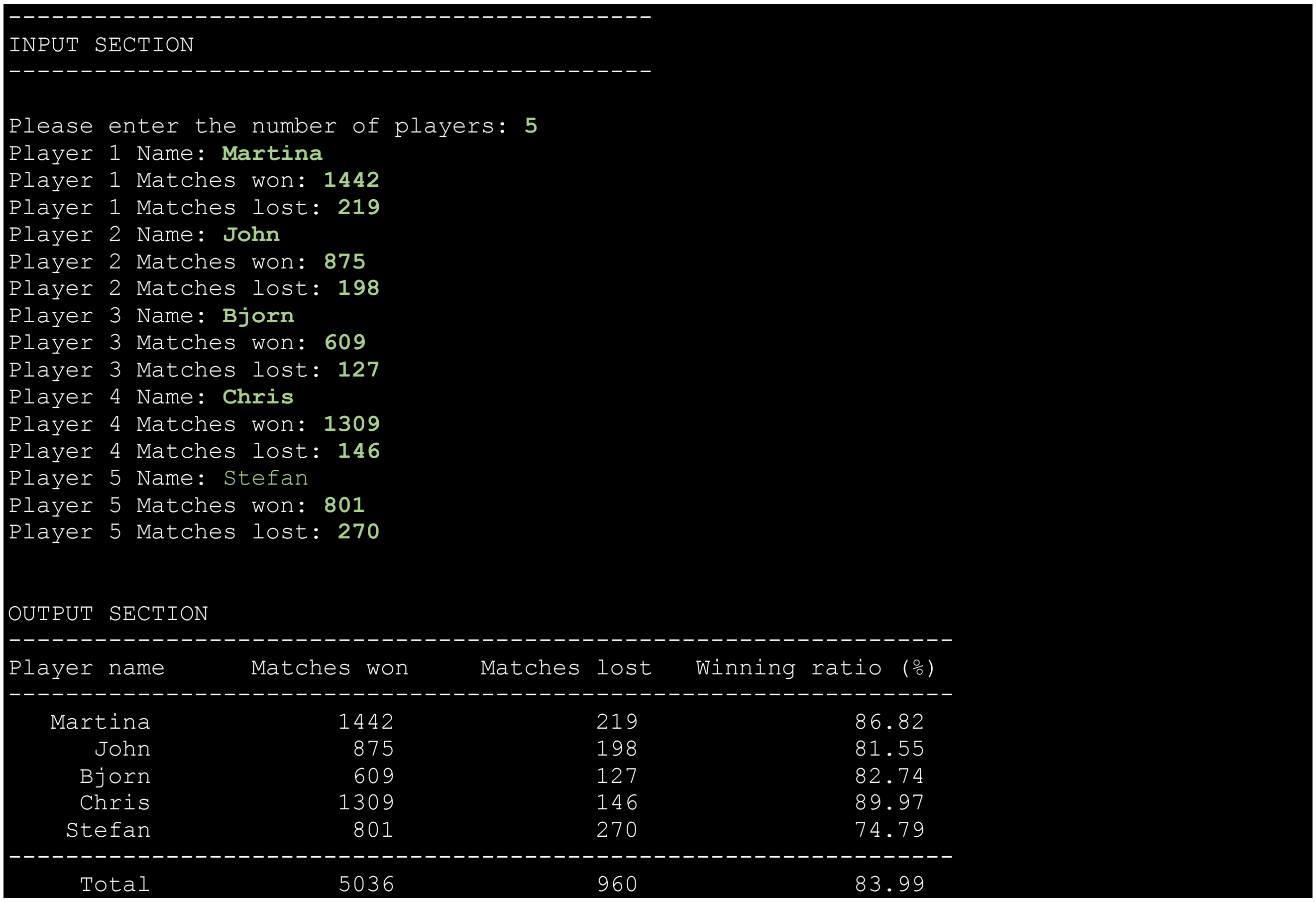
SQ 2.2

Write a program that:

- asks the user for the number of records for tennis players, N
- uses one array of Strings of length N to store track player names, an array of integers of length N to store the matches won, and an array of integers to store the matches lost.
- reads a list of N player names, the number of matches won, and the number of matches lost, and stores them in the corresponding arrays
- calculates the winning ratio (in percent) for each player as:
    - `100×Matcheswon/TotalnumberOfMatchesPlayed`

- keeps track of the total number of matches won by each player, and the total number of matches lost by them
- prints all N player names, matches won, matches lost, and winning ratios
- At the end, prints the total matches won, total matches lost, and the overall winning ratio

Sample Run



```
---------------------------------------------
INPUT SECTION
---------------------------------------------

Please enter the number of players: 5
Player 1 Name: Martina
Player 1 Matches won: 1442
Player 1 Matches lost: 219
Player 2 Name: John
Player 2 Matches won: 875
Player 2 Matches lost: 198
Player 3 Name: Bjorn
Player 3 Matches won: 609
Player 3 Matches lost: 127
Player 4 Name: Chris
Player 4 Matches won: 1309
Player 4 Matches lost: 146
Player 5 Name: Stefan
Player 5 Matches won: 801
Player 5 Matches lost: 270


OUTPUT SECTION
------------------------------------------------------------------
Player name      Matches won     Matches lost   Winning ratio (%)
------------------------------------------------------------------
   Martina             1442             219               86.82
      John              875             198               81.55
     Bjorn              609             127               82.74
     Chris             1309             146               89.97
    Stefan              801             270               74.79
------------------------------------------------------------------
     Total             5036             960               83.99
```

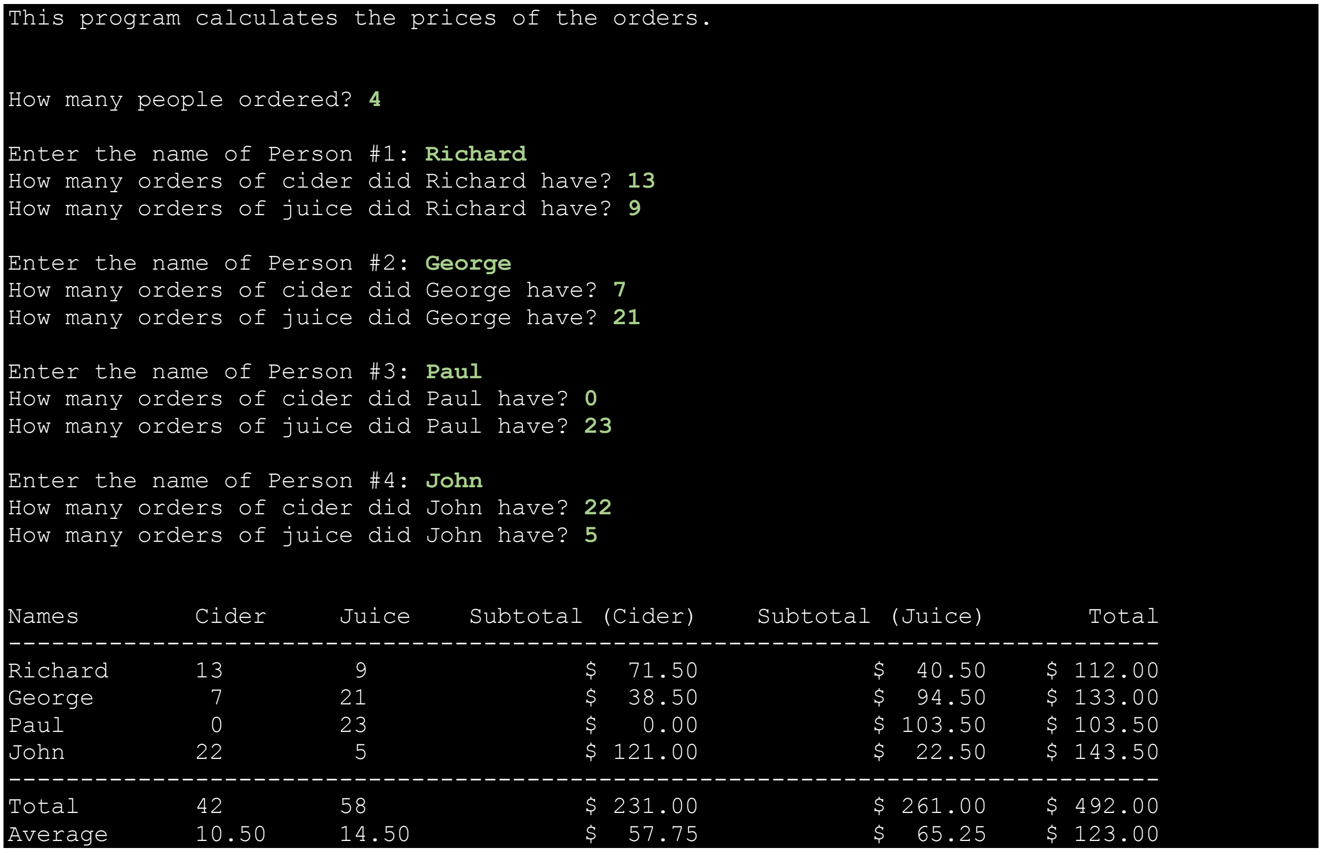
SQ 2.3

Write a program that calculates the prices of the orders (apple cider and apple juice). Your program should:

- ask for the number of people who ordered
- use an array of Strings to store names, an array of integers to store the number of orders of apple cider (each priced at $5.50), and an array of integers to store the number of orders of apple juice (each priced at $4.50)
- read a list of names, numbers of apple cider orders, and numbers of apple juice
- store them in the corresponding arrays
- keep track of the total number of apple cider orders and the total number of apple juice orders
- print all the names, the numbers of orders for each kind, the subtotal prices for each kind, and the total price for each person
- At the end, it should also print:
- the total number of orders for each kind, the subtotal prices of the total orders, and the grand total price
- an average number of orders for each kind, an average subtotal price of each kind, and an average price for each person

Do not worry about having exactly two decimal points and other fancy formatting.

Sample Run



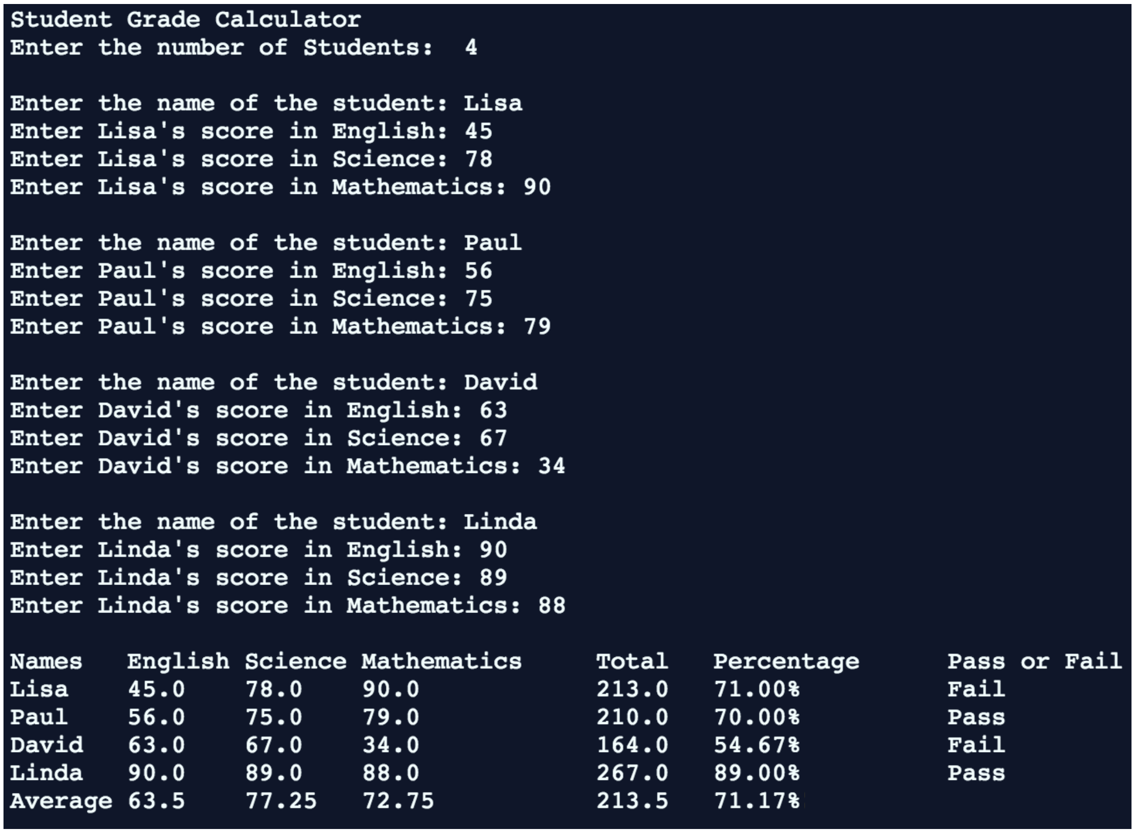
```
This program calculates the prices of the orders.


How many people ordered? 4

Enter the name of Person #1: Richard
How many orders of cider did Richard have? 13
How many orders of juice did Richard have? 9

Enter the name of Person #2: George
How many orders of cider did George have? 7
How many orders of juice did George have? 21

Enter the name of Person #3: Paul
How many orders of cider did Paul have? 0
How many orders of juice did Paul have? 23

Enter the name of Person #4: John
How many orders of cider did John have? 22
How many orders of juice did John have? 5


Names        Cider     Juice    Subtotal (Cider)    Subtotal (Juice)       Total
--------------------------------------------------------------------------------
Richard      13         9               $  71.50            $  40.50    $ 112.00
George        7        21               $  38.50            $  94.50    $ 133.00
Paul          0        23               $   0.00            $ 103.50    $ 103.50
John         22         5               $ 121.00            $  22.50    $ 143.50
--------------------------------------------------------------------------------
Total        42        58               $ 231.00            $ 261.00    $ 492.00
Average      10.50     14.50            $  57.75            $  65.25    $ 123.00
```

SQ 2.4

Write a program that will:

- read the number of students N
- use a couple of arrays of length N to store student names and scores obtained for each course - English, Mathematics, and Science (marks range from 0-100)
- read a list of N student names, and scores obtained in each course, and store them in the corresponding arrays
- calculate the total score for each student
- calculate the corresponding percentage (percentage = total/3)
- keep track of the student grades based on the two conditions:
    - If the score obtained by the student in any of the courses (English, Mathematics and Science) is less than 50 then mark the student as a Fail
    - If the scores in all the courses (English, Mathematics and Science) are 50 and above, then mark the student as Pass
- print all the student names, the scores obtained in all the courses, their corresponding total, percentage, and pass or fail in a tabular format
- and at the end, print the overall average score for each course, their Total and Percentage values.
- Do not worry about having exact decimal points and other number formatting.

Sample Run



```
Student Grade Calculator
Enter the number of Students:  4

Enter the name of the student: Lisa
Enter Lisa's score in English: 45
Enter Lisa's score in Science: 78
Enter Lisa's score in Mathematics: 90

Enter the name of the student: Paul
Enter Paul's score in English: 56
Enter Paul's score in Science: 75
Enter Paul's score in Mathematics: 79

Enter the name of the student: David
Enter David's score in English: 63
Enter David's score in Science: 67
Enter David's score in Mathematics: 34

Enter the name of the student: Linda
Enter Linda's score in English: 90
Enter Linda's score in Science: 89
Enter Linda's score in Mathematics: 88

Names   English Science Mathematics     Total   Percentage      Pass or Fail
Lisa    45.0    78.0    90.0            213.0   71.00%          Fail
Paul    56.0    75.0    79.0            210.0   70.00%          Pass
David   63.0    67.0    34.0            164.0   54.67%          Fail
Linda   90.0    89.0    88.0            267.0   89.00%          Pass
Average 63.5    77.25   72.75           213.5   71.17%
```

SQ 2.5

Write a program to calculate the employee's salaries based on their yearly performance.

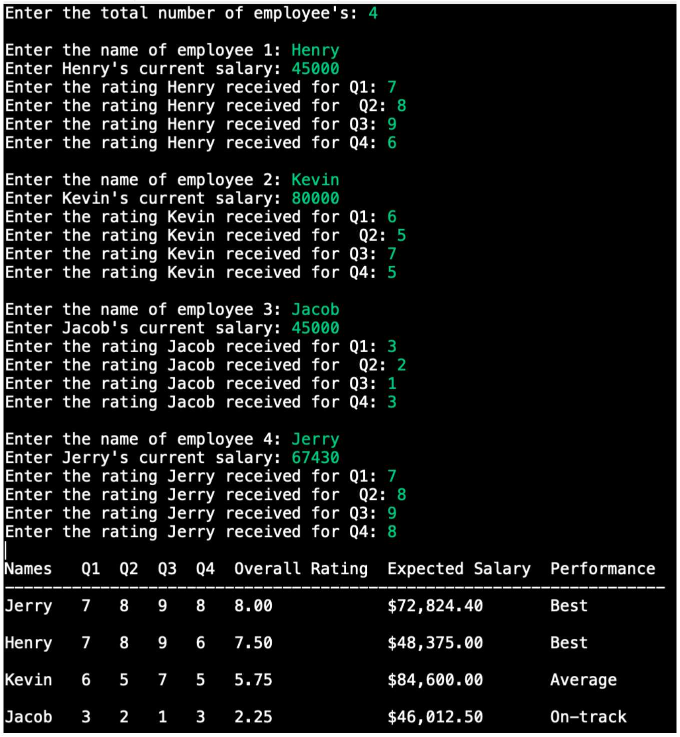
The program should:

- Read the number of employees N;
- Use a few arrays of length N to store employee names, their current salaries, and their ratings obtained for each quarter - Q1, Q2, Q3, and Q4 (ratings ranging from 1-10);
- Read a list of N employee names, current salaries, and quarterly ratings, and store them in the corresponding arrays;
- Calculate the overall (i.e., average) rating for each employee for the year (total rating/4);
- Calculate the expected salary for the next year based on the overall performance;
- increase in salary next year = current salary *(overall rating/100)
- Expected salary = current salary + amount increased
- Keep track of the employee's performance based on the three conditions:
    - If the employee's overall rating is greater than or equal to 7 then the performance is tagged "BEST".
    - If the employee's overall rating is greater than or equal to 5 and less than 7 then the performance is tagged "AVERAGE".
    - If the employee's overall rating is less than 5 then the performance is tagged "ON-TRACK".
- Sort the records based on the overall rating in descending order, starting with the best performance to the on-track performance; and,

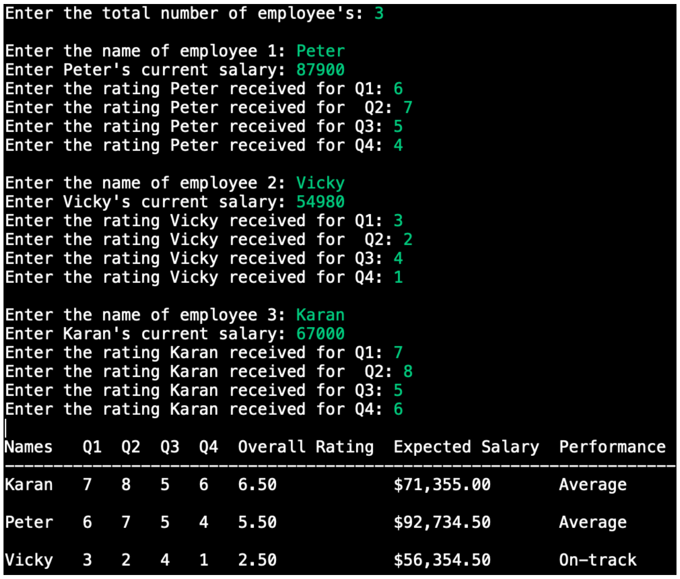
- Print all the employee names, the ratings obtained from Q1 through Q4, the overall ratings, and the performance indicators in a tabular format.

You do not need to worry about having exactly two decimal points and other fancy number formatting.

Sample Runs



```
Enter the total number of employee's: 4

Enter the name of employee 1: Henry
Enter Henry's current salary: 45000
Enter the rating Henry received for Q1: 7
Enter the rating Henry received for  Q2: 8
Enter the rating Henry received for Q3: 9
Enter the rating Henry received for Q4: 6

Enter the name of employee 2: Kevin
Enter Kevin's current salary: 80000
Enter the rating Kevin received for Q1: 6
Enter the rating Kevin received for  Q2: 5
Enter the rating Kevin received for Q3: 7
Enter the rating Kevin received for Q4: 5

Enter the name of employee 3: Jacob
Enter Jacob's current salary: 45000
Enter the rating Jacob received for Q1: 3
Enter the rating Jacob received for  Q2: 2
Enter the rating Jacob received for Q3: 1
Enter the rating Jacob received for Q4: 3

Enter the name of employee 4: Jerry
Enter Jerry's current salary: 67430
Enter the rating Jerry received for Q1: 7
Enter the rating Jerry received for  Q2: 8
Enter the rating Jerry received for Q3: 9
Enter the rating Jerry received for Q4: 8

Names   Q1  Q2  Q3  Q4  Overall Rating  Expected Salary  Performance
----------------------------------------------------------------------
Jerry   7   8   9   8   8.00            $72,824.40       Best

Henry   7   8   9   6   7.50            $48,375.00       Best

Kevin   6   5   7   5   5.75            $84,600.00       Average

Jacob   3   2   1   3   2.25            $46,012.50       On-track
```



```
Enter the total number of employee's: 3

Enter the name of employee 1: Peter
Enter Peter's current salary: 87900
Enter the rating Peter received for Q1: 6
Enter the rating Peter received for  Q2: 7
Enter the rating Peter received for Q3: 5
Enter the rating Peter received for Q4: 4

Enter the name of employee 2: Vicky
Enter Vicky's current salary: 54980
Enter the rating Vicky received for Q1: 3
Enter the rating Vicky received for  Q2: 2
Enter the rating Vicky received for Q3: 4
Enter the rating Vicky received for Q4: 1

Enter the name of employee 3: Karan
Enter Karan's current salary: 67000
Enter the rating Karan received for Q1: 7
Enter the rating Karan received for  Q2: 8
Enter the rating Karan received for Q3: 5
Enter the rating Karan received for Q4: 6

Names   Q1  Q2  Q3  Q4  Overall Rating  Expected Salary  Performance
----------------------------------------------------------------------
Karan   7   8   5   6   6.50            $71,355.00       Average

Peter   6   7   5   4   5.50            $92,734.50       Average

Vicky   3   2   4   1   2.50            $56,354.50       On-track
```

SQ 2.6

Write a program to calculate the profit/loss of *N* products based on their purchase costs, selling prices, and monthly sales.

The program should:

- Read the number of products N.
- Use arrays of length N to store the following:
    - the product names
    - the number of purchases of each product
    - the number of sales made for each product
    - their purchase cost
    - their selling prices
    - the profit or loss incurred from each product
- Calculate the product profit or loss percentage based on the purchase cost and the sales made:

```
<Product profit/loss percentage>
            = (<Product Profit or Loss> / <Total product purchase cost>) * 100
```

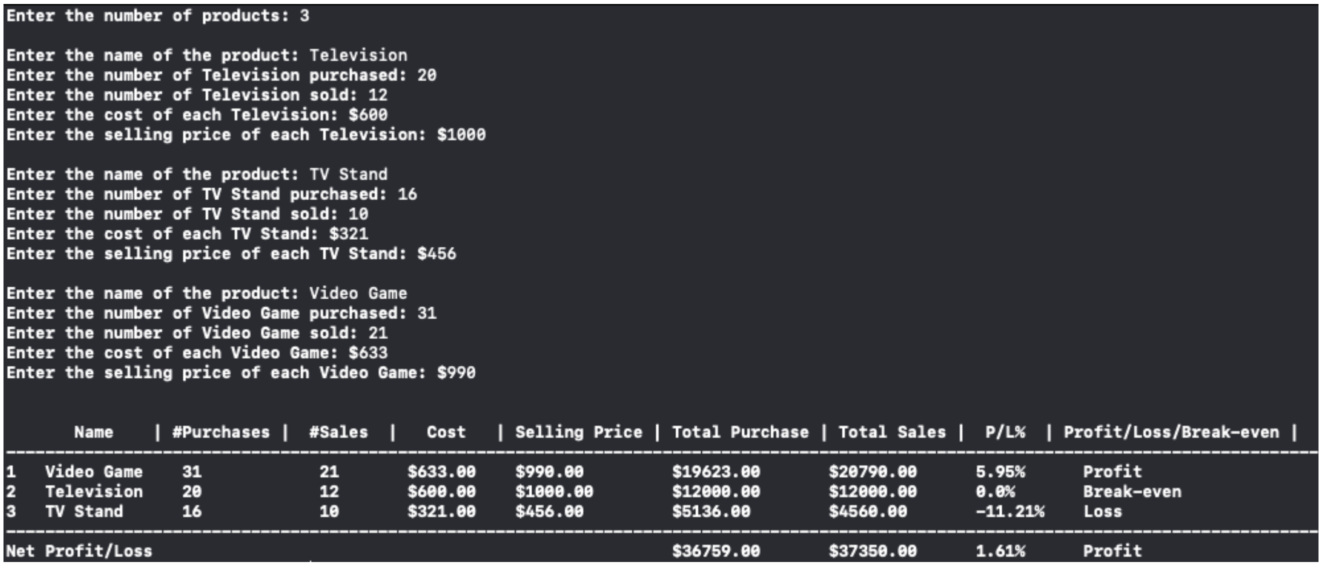
where,

```
<Product Profit or Loss>
              =  <Total product sales> - <Total product purchase cost>
<Total product purchase cost>
              = <purchase cost of each product> * <number of products purchased>
<Total product sales>
              = <selling price of each product> * <number of products sold>
```

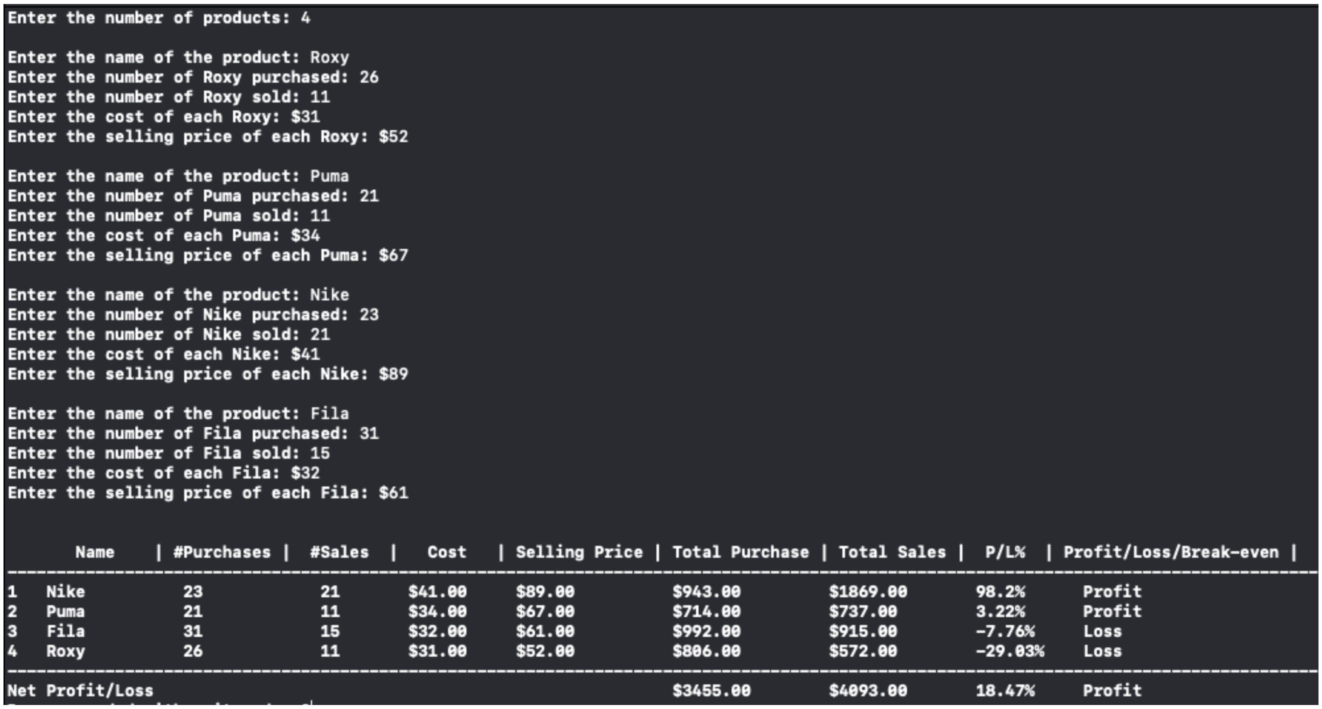
- Also, calculate the net profit or loss percentage based on the net profit/loss and the net purchase cost.
- Keep track of the product profit/loss based on the three conditions:
  - If the profit/loss percentage is greater than 0 then mark the product sales as “PROFIT”.
  - If the profit/loss percentage is equal to 0 then mark the product sales as “BREAK-EVEN”.
  - If the profit/loss percentage is less than 0 then mark the product sales as “LOSS”.
- Sort the products in descending order based on the profit/loss percentage of each product, starting with the most profitable product to the least.
- For each product, print (in a tabular format): the product name, the number of purchases, the number of sales, the purchase cost, the selling price, the total purchase cost, the total sales, the overall profit/loss percentage, and the sales indicators.
- At the end, print the net purchase cost, the net sales, and the net profit/loss percentage and the profit/loss indicator.

You do not need to worry about having exactly two decimal points and other fancy number formatting.

Sample Runs



```
Enter the number of products: 3

Enter the name of the product: Television
Enter the number of Television purchased: 20
Enter the number of Television sold: 12
Enter the cost of each Television: $600
Enter the selling price of each Television: $1000

Enter the name of the product: TV Stand
Enter the number of TV Stand purchased: 16
Enter the number of TV Stand sold: 10
Enter the cost of each TV Stand: $321
Enter the selling price of each TV Stand: $456

Enter the name of the product: Video Game
Enter the number of Video Game purchased: 31
Enter the number of Video Game sold: 21
Enter the cost of each Video Game: $633
Enter the selling price of each Video Game: $990


       Name    | #Purchases |  #Sales  |   Cost   | Selling Price | Total Purchase | Total Sales |  P/L%  | Profit/Loss/Break-even |
---------------------------------------------------------------------------------------------------------------------------------------
1   Video Game    31            21       $633.00    $990.00         $19623.00       $20790.00      5.95%      Profit
2   Television    20            12       $600.00    $1000.00        $12000.00       $12000.00      0.0%       Break-even
3   TV Stand      16            10       $321.00    $456.00         $5136.00        $4560.00       -11.21%    Loss
---------------------------------------------------------------------------------------------------------------------------------------
Net Profit/Loss                                                     $36759.00       $37350.00      1.61%      Profit
```



```
Enter the number of products: 4

Enter the name of the product: Roxy
Enter the number of Roxy purchased: 26
Enter the number of Roxy sold: 11
Enter the cost of each Roxy: $31
Enter the selling price of each Roxy: $52

Enter the name of the product: Puma
Enter the number of Puma purchased: 21
Enter the number of Puma sold: 11
Enter the cost of each Puma: $34
Enter the selling price of each Puma: $67

Enter the name of the product: Nike
Enter the number of Nike purchased: 23
Enter the number of Nike sold: 21
Enter the cost of each Nike: $41
Enter the selling price of each Nike: $89

Enter the name of the product: Fila
Enter the number of Fila purchased: 31
Enter the number of Fila sold: 15
Enter the cost of each Fila: $32
Enter the selling price of each Fila: $61


       Name     | #Purchases |  #Sales  |   Cost    | Selling Price | Total Purchase | Total Sales |  P/L%  | Profit/Loss/Break-even |
-------------------------------------------------------------------------------------------------------------------------------------
1   Nike          23            21       $41.00     $89.00           $943.00          $1869.00      98.2%      Profit
2   Puma          21            11       $34.00     $67.00           $714.00          $737.00       3.22%      Profit
3   Fila          31            15       $32.00     $61.00           $992.00          $915.00       -7.76%     Loss
4   Roxy          26            11       $31.00     $52.00           $806.00          $572.00       -29.03%    Loss
-------------------------------------------------------------------------------------------------------------------------------------
Net Profit/Loss                                                      $3455.00         $4093.00      18.47%     Profit
```

SQ 2.7

Write a program to recalculate the mortgage repayment of n customers based on their mortgage balance, interest rate, current monthly payment, and extra monthly payment that they plan to increase the current payment by.

The program should:

- Read the number of customers n
- Use arrays of length n to store the following:
  - the borrower names
  - the mortgage balances
  - the interest rates
  - the current monthly payments
  - the extra monthly payments
- For each customer, calculate the current and new: monthly payment, duration, and interest amount
  i.e.)
  <New Payment> = <Current Payment> + <Extra Payment>
  <Current Duration (in months)> =

$$\frac{\log\left[\frac{\frac{PMT}{i}}{\frac{PMT}{i}-PV}\right]}{\log(1+i)}$$

PMT = <Current payment>
i = <Interest rate> / 100 / 12
PV = <Mortgage balance>

(Hint: Round the current duration to the closest integer)*

<New Duration (in months)> =

$$\frac{\log\left[\frac{\frac{PMT}{i}}{\frac{PMT}{i}-PV}\right]}{\log(1+i)}$$

PMT = <New payment>
i = <Interest rate> / 100 / 12
PV = <Mortgage balance>

(Hint: Round the new duration to the closest integer)*

<Current Interest>
= (<Current Payment> <Current Duration (in months)>) - <Mortgage balance>*

<New Interest>
= (<New Payment> <New Duration (in months)>) - <Mortgage balance>*

<Current Duration (in years & months)>

$$= \begin{cases} \textit{<Current Duration (in months)> / 12 (years)} \\ \textit{<Current Duration (in months)> \% 12 (months)} \end{cases}$$

< New Duration (in years & months)> =

$$= \begin{cases} \textit{<New Duration (in months)> / 12 (years)} \\ \textit{<New Duration (in months)> \% 12 (months)} \end{cases}$$

(Example: 125 months = 125/12= 10 (years) & 125%12 = 5 months)*

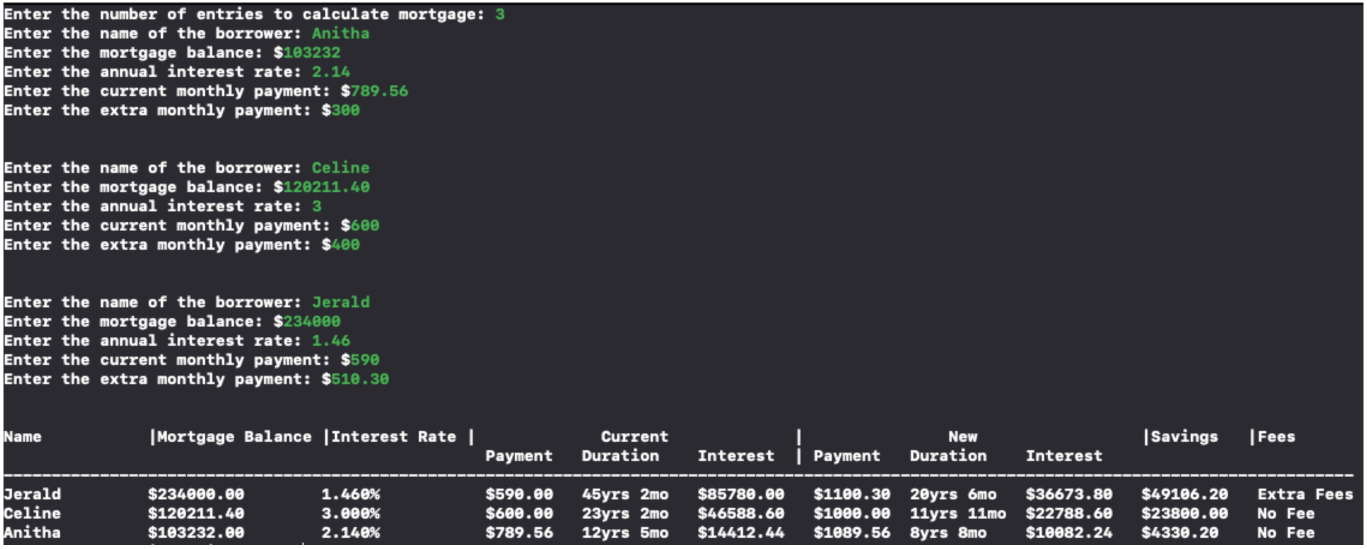
- Calculate the savings on each mortgage based on the interest calculated
  <Savings> = <Current Interest> - <New Interest>
- Keep track of the fees based on two conditions:
  - If the new duration is **less than or equal to half of the current duration**, then mark the mortgage with an "**Extra Fees**".
  - Otherwise, mark the mortgage with a "**No Fee**".
- Sort the mortgages in descending order based on the overall savings, starting with the highest savings incurred on a mortgage to the least savings incurred on a mortgage.

Print the following:

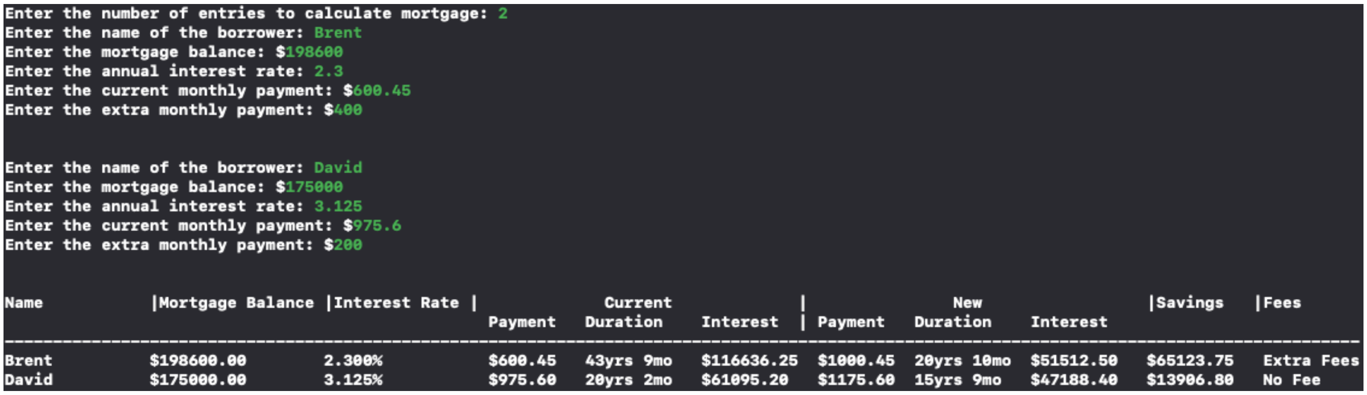
- all the borrower names
- their mortgage balances
- their interest rates
- their current: [payments, durations (in years/months), interest amounts]
- their new: [payments, durations (in years/months), interest amounts]
- their savings

And indicate whether there are extra fees (Extra Fees/No Fee).

Sample Runs



```
Enter the number of entries to calculate mortgage: 3
Enter the name of the borrower: Anitha
Enter the mortgage balance: $103232
Enter the annual interest rate: 2.14
Enter the current monthly payment: $789.56
Enter the extra monthly payment: $300


Enter the name of the borrower: Celine
Enter the mortgage balance: $120211.40
Enter the annual interest rate: 3
Enter the current monthly payment: $600
Enter the extra monthly payment: $400


Enter the name of the borrower: Jerald
Enter the mortgage balance: $234000
Enter the annual interest rate: 1.46
Enter the current monthly payment: $590
Enter the extra monthly payment: $510.30


Name            |Mortgage Balance |Interest Rate |            Current                |              New                    |Savings    |Fees
                                                  Payment    Duration     Interest   | Payment    Duration     Interest
--------------------------------------------------------------------------------------------------------------------------------------------
Jerald          $234000.00        1.460%          $590.00    45yrs 2mo    $85780.00   $1100.30   20yrs 6mo    $36673.80   $49106.20   Extra Fees
Celine          $120211.40        3.000%          $600.00    23yrs 2mo    $46588.60   $1000.00   11yrs 11mo   $22788.60   $23800.00   No Fee
Anitha          $103232.00        2.140%          $789.56    12yrs 5mo    $14412.44   $1089.56   8yrs 8mo     $10082.24   $4330.20    No Fee
```



```
Enter the number of entries to calculate mortgage: 2
Enter the name of the borrower: Brent
Enter the mortgage balance: $198600
Enter the annual interest rate: 2.3
Enter the current monthly payment: $600.45
Enter the extra monthly payment: $400


Enter the name of the borrower: David
Enter the mortgage balance: $175000
Enter the annual interest rate: 3.125
Enter the current monthly payment: $975.6
Enter the extra monthly payment: $200


Name            |Mortgage Balance |Interest Rate |            Current                |              New                    |Savings    |Fees
                                                  Payment    Duration     Interest   | Payment    Duration     Interest
--------------------------------------------------------------------------------------------------------------------------------------------
Brent           $198600.00        2.300%          $600.45    43yrs 9mo    $116636.25 $1000.45   20yrs 10mo   $51512.50   $65123.75   Extra Fees
David           $175000.00        3.125%          $975.60    20yrs 2mo    $61095.20   $1175.60   15yrs 9mo    $47188.40   $13906.80   No Fee
```

SQ 2.8

Write a program to determine which project should be executed using cost-benefit analysis.

If the upfront cost is incurred, using the cash flow during a single period and the discount rate, calculate the Net Present Value (NPV) of the project. Also, determine whether the project is viable by comparing projects based on their anticipated revenue and NPV value.

The program should:

- Read the number of projects n
- Use arrays of length n to store the following:
    - the name of the project
    - the upfront cost
    - the rate of return (in %)
    - the duration (in years)
    - an array of cash inflows-outflows i.e., of length of duration d
- For each project and corresponding year, calculate the PV factor and the amount given in each year's annual cash flow.

i.e.)

<PV Factor> = $\dfrac{1}{(1+r)^{n}}$

where:

r = <Rate of return or discount rate> / 100

n = <Duration of year(s)>

(Hint: Round the PV Factor to four decimal places)*

*<Amount> = <Cash Inflow/Outflows> * <PV Factor>*

- For each project, calculate the total income, present value of future benefits, and net present value.

  <Total Income> = <Sum of "Cash Inflow/Outflows">
  <Present Value of Future Benefits> = <Sum of "Amount">
  <Present Value of Future Costs> = Upfront Cost
  <Net Present Value> = <Present Value of Future Benefits> − <Present Value of Future Costs>
  (Example: Net Present Value For Project Mars = 43976.63 – 35000.00 = 8976.63)*

  (Hint: Round all the dollar amounts to two decimal places)*

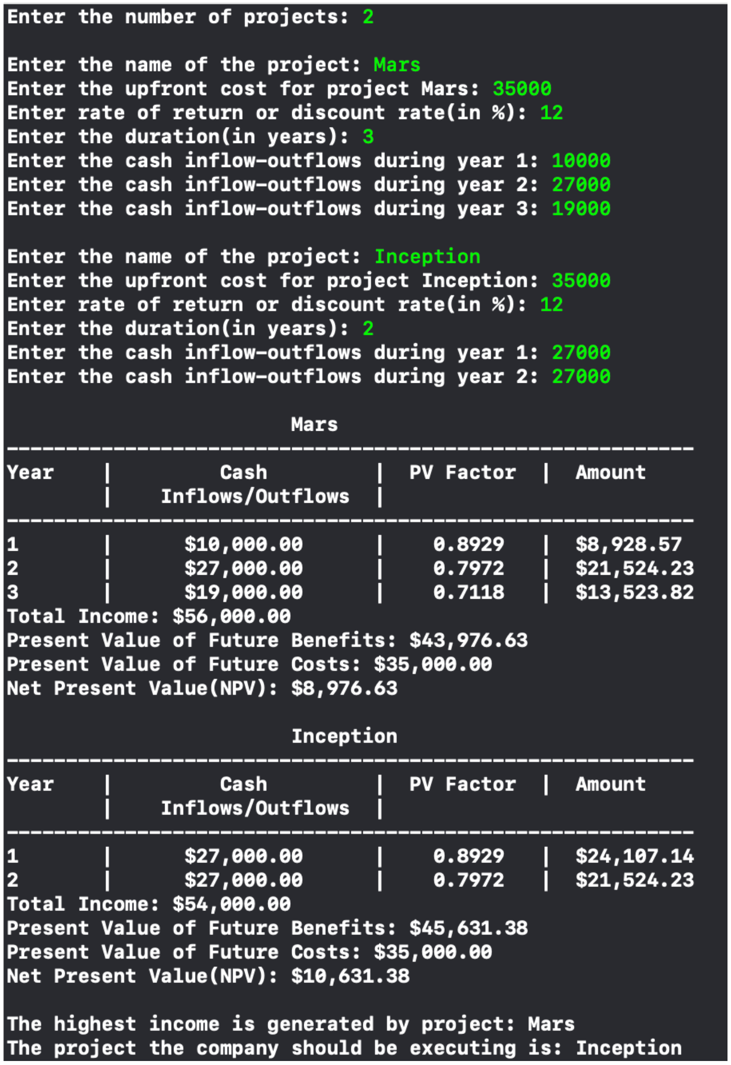
Print the following for each project:

- the project name
- for a specific year; the cash inflows/outflows, PV factor, and amount
- the total income
- the present value of future benefits
- the present value of future costs
- the net present value (NPV)

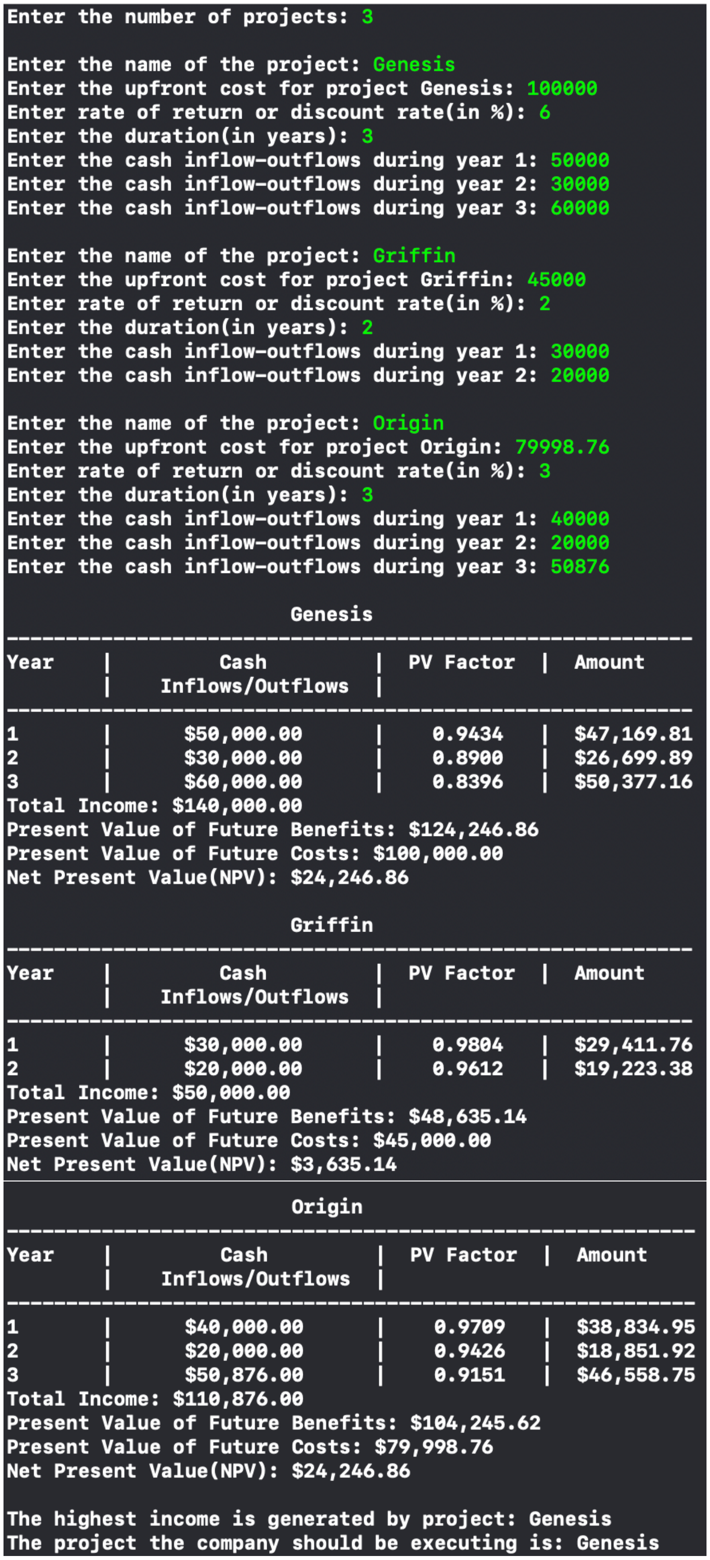
At the end:

- Compare the total income and indicate the project that generates the **maximum total income** with the following condition:
  - ➢ If any two or more projects' total incomes are equal, then the project with the highest NPV is considered as generating the highest income.
- Indicate the project the company should be executing based on the NPV value. Typically, projects with the **highest NPV** are pursued with the following condition:
  - ➢ If any two or more projects' NPV are **equal**, then the project with the **maximum Total Income** should be executed.

Sample Runs



```
Enter the number of projects: 2

Enter the name of the project: Mars
Enter the upfront cost for project Mars: 35000
Enter rate of return or discount rate(in %): 12
Enter the duration(in years): 3
Enter the cash inflow-outflows during year 1: 10000
Enter the cash inflow-outflows during year 2: 27000
Enter the cash inflow-outflows during year 3: 19000

Enter the name of the project: Inception
Enter the upfront cost for project Inception: 35000
Enter rate of return or discount rate(in %): 12
Enter the duration(in years): 2
Enter the cash inflow-outflows during year 1: 27000
Enter the cash inflow-outflows during year 2: 27000

                        Mars
----------------------------------------------------------
Year    |         Cash         |  PV Factor  |  Amount
        |    Inflows/Outflows  |
----------------------------------------------------------
1       |      $10,000.00      |    0.8929   |  $8,928.57
2       |      $27,000.00      |    0.7972   |  $21,524.23
3       |      $19,000.00      |    0.7118   |  $13,523.82
Total Income: $56,000.00
Present Value of Future Benefits: $43,976.63
Present Value of Future Costs: $35,000.00
Net Present Value(NPV): $8,976.63

                        Inception
----------------------------------------------------------
Year    |         Cash         |  PV Factor  |  Amount
        |    Inflows/Outflows  |
----------------------------------------------------------
1       |      $27,000.00      |    0.8929   |  $24,107.14
2       |      $27,000.00      |    0.7972   |  $21,524.23
Total Income: $54,000.00
Present Value of Future Benefits: $45,631.38
Present Value of Future Costs: $35,000.00
Net Present Value(NPV): $10,631.38

The highest income is generated by project: Mars
The project the company should be executing is: Inception
```



```
Enter the number of projects: 3

Enter the name of the project: Genesis
Enter the upfront cost for project Genesis: 100000
Enter rate of return or discount rate(in %): 6
Enter the duration(in years): 3
Enter the cash inflow-outflows during year 1: 50000
Enter the cash inflow-outflows during year 2: 30000
Enter the cash inflow-outflows during year 3: 60000

Enter the name of the project: Griffin
Enter the upfront cost for project Griffin: 45000
Enter rate of return or discount rate(in %): 2
Enter the duration(in years): 2
Enter the cash inflow-outflows during year 1: 30000
Enter the cash inflow-outflows during year 2: 20000

Enter the name of the project: Origin
Enter the upfront cost for project Origin: 79998.76
Enter rate of return or discount rate(in %): 3
Enter the duration(in years): 3
Enter the cash inflow-outflows during year 1: 40000
Enter the cash inflow-outflows during year 2: 20000
Enter the cash inflow-outflows during year 3: 50876

                        Genesis
----------------------------------------------------------
Year    |         Cash          |  PV Factor  |  Amount
        |    Inflows/Outflows   |
----------------------------------------------------------
1       |      $50,000.00       |    0.9434   |  $47,169.81
2       |      $30,000.00       |    0.8900   |  $26,699.89
3       |      $60,000.00       |    0.8396   |  $50,377.16
Total Income: $140,000.00
Present Value of Future Benefits: $124,246.86
Present Value of Future Costs: $100,000.00
Net Present Value(NPV): $24,246.86

                        Griffin
----------------------------------------------------------
Year    |         Cash          |  PV Factor  |  Amount
        |    Inflows/Outflows   |
----------------------------------------------------------
1       |      $30,000.00       |    0.9804   |  $29,411.76
2       |      $20,000.00       |    0.9612   |  $19,223.38
Total Income: $50,000.00
Present Value of Future Benefits: $48,635.14
Present Value of Future Costs: $45,000.00
Net Present Value(NPV): $3,635.14

                        Origin
----------------------------------------------------------
Year    |         Cash          |  PV Factor  |  Amount
        |    Inflows/Outflows   |
----------------------------------------------------------
1       |      $40,000.00       |    0.9709   |  $38,834.95
2       |      $20,000.00       |    0.9426   |  $18,851.92
3       |      $50,876.00       |    0.9151   |  $46,558.75
Total Income: $110,876.00
Present Value of Future Benefits: $104,245.62
Present Value of Future Costs: $79,998.76
Net Present Value(NPV): $24,246.86

The highest income is generated by project: Genesis
The project the company should be executing is: Genesis
```

SQ 2.9

Write a program to measure the economy of countries by calculating their Growth Rate.

If, for a given period, the GDP value at the beginning and subsequent years is known, calculate the Growth Rate (GR) of the country for each year, the Average Annual Growth Rate (AAGR) and the Compound Annual Growth Rate (CAGR). Also, measure the relative riskiness of the country's economy based on its standard deviation.

The program should:

- Read the number of countries a
- Read the number of years n
- Use an array of length a to store the following:
  - the name of the country
  - an array of GDP i.e., of length of years n
  - an array of GR values i.e., of length of years n
- For each country's GDP value and corresponding year, calculate the growth rate (GR) and its ranking.

  i.e.)

  <Growth Rate (GR)> = $\dfrac{(EV-BV)}{BV}$

  where:
  EV (Ending Value) = <GDP Value of the current year >
  BV (Beginning Value) = < GDP Value of the previous year>

  (Hint: Round the GR value to two decimal places)*

  <Ranking> = GR > 25 = ***Exceptional***
  GR > 0 = ***Good***
  GR < 0 = ***Poor***

- For each country calculate the average annual growth rate, compound annual growth rate and standard deviation.

  <Average Annual Growth Rate> = <Sum of "Growth Rate" / n>
  < Compound Annual Growth Rate > =

$$\left(\frac{EV}{BV}\right)^{\frac{1}{n}} - 1$$

  where:
  EV = <GDP Value of the final year >
  BV = < GDP Value of the beginning year>

  <Standard Deviation> =

$$\sqrt{\frac{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2}{n-1}}$$

  where:
  $\overline{x}$ *= <Mean of GR values>*
  x_i *= < GR Value of each year>*

(Hint: Round the GR, AAGR and CAGR to two decimal places and the standard deviation to three decimal places)*

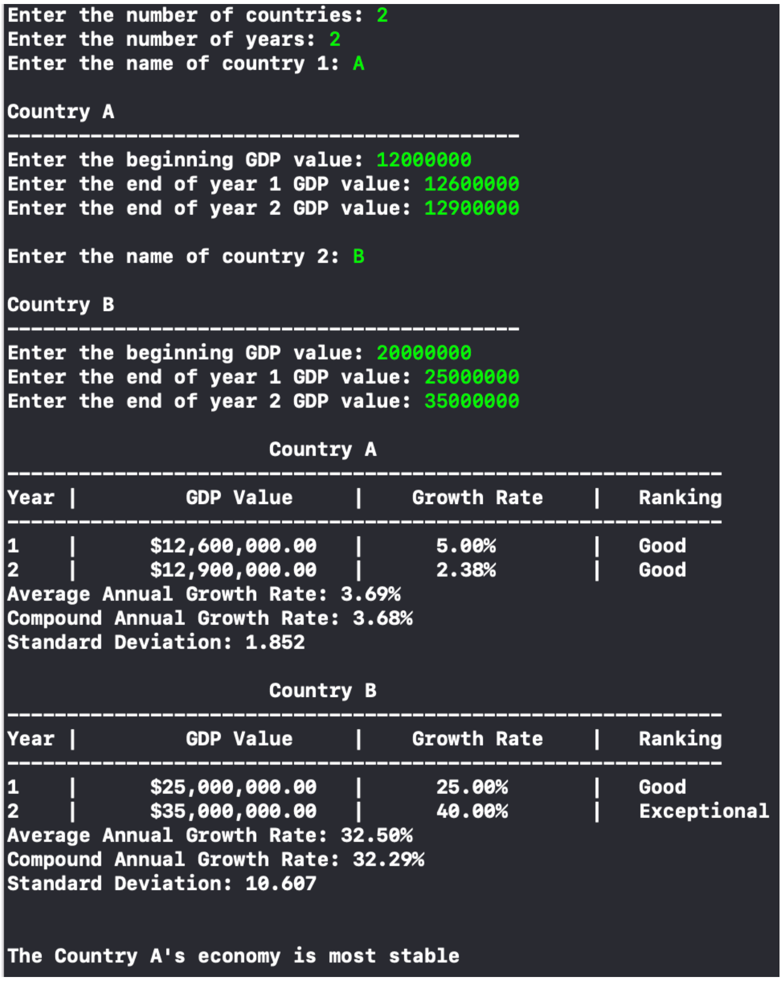
Print the following for each country:

- the country name
- for a specific year; the GDP Value, Growth Rate and Ranking
- the average annual growth rate
- the compound annual growth rate
- the standard deviation

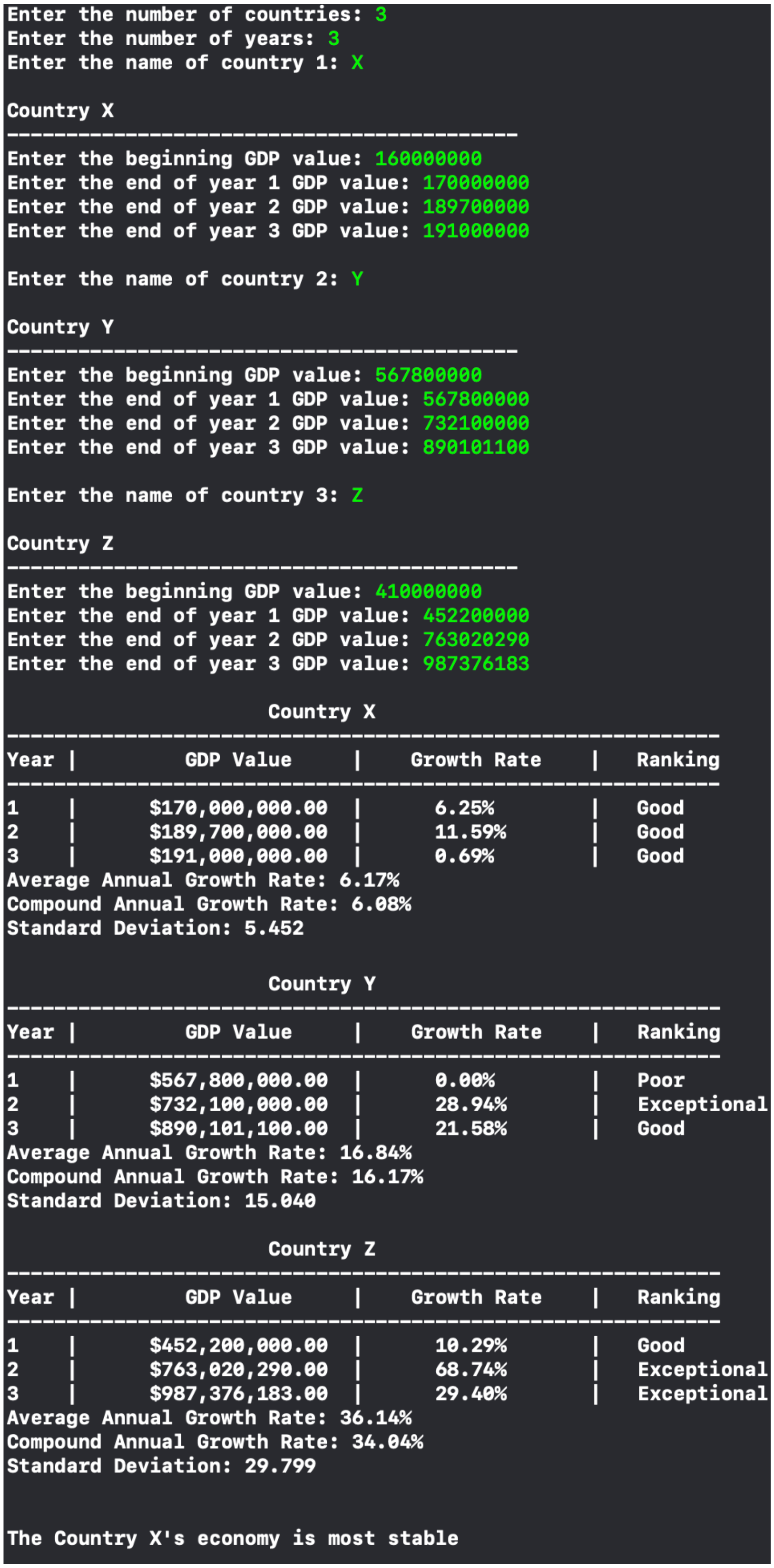
At the end:

- Compare the standard deviation of each country to measure the economic volatility based on the **minimum standard deviation** with the following condition:
    - ➢ If **minimum standard deviation < 25**, then print;
    - “The Country <country name>'s economy is most stable”
    - ➢ If **minimum standard deviation > 25**, then print;
    - “The Country <country name>'s economy is least risky”

<u>Sample Runs</u>



```
Enter the number of countries: 2
Enter the number of years: 2
Enter the name of country 1: A

Country A
-------------------------------------------
Enter the beginning GDP value: 12000000
Enter the end of year 1 GDP value: 12600000
Enter the end of year 2 GDP value: 12900000

Enter the name of country 2: B

Country B
-------------------------------------------
Enter the beginning GDP value: 20000000
Enter the end of year 1 GDP value: 25000000
Enter the end of year 2 GDP value: 35000000

                      Country A
------------------------------------------------------------
Year |         GDP Value     |    Growth Rate     |   Ranking
------------------------------------------------------------
1    |      $12,600,000.00   |      5.00%         |   Good
2    |      $12,900,000.00   |      2.38%         |   Good
Average Annual Growth Rate: 3.69%
Compound Annual Growth Rate: 3.68%
Standard Deviation: 1.852

                      Country B
------------------------------------------------------------
Year |         GDP Value     |    Growth Rate     |   Ranking
------------------------------------------------------------
1    |      $25,000,000.00   |      25.00%        |   Good
2    |      $35,000,000.00   |      40.00%        |   Exceptional
Average Annual Growth Rate: 32.50%
Compound Annual Growth Rate: 32.29%
Standard Deviation: 10.607


The Country A's economy is most stable
```



```
Enter the number of countries: 3
Enter the number of years: 3
Enter the name of country 1: X

Country X
-------------------------------------------
Enter the beginning GDP value: 160000000
Enter the end of year 1 GDP value: 170000000
Enter the end of year 2 GDP value: 189700000
Enter the end of year 3 GDP value: 191000000

Enter the name of country 2: Y

Country Y
-------------------------------------------
Enter the beginning GDP value: 567800000
Enter the end of year 1 GDP value: 567800000
Enter the end of year 2 GDP value: 732100000
Enter the end of year 3 GDP value: 890101100

Enter the name of country 3: Z

Country Z
-------------------------------------------
Enter the beginning GDP value: 410000000
Enter the end of year 1 GDP value: 452200000
Enter the end of year 2 GDP value: 763020290
Enter the end of year 3 GDP value: 987376183

                      Country X
------------------------------------------------------------
Year |         GDP Value      |    Growth Rate     |   Ranking
------------------------------------------------------------
1    |      $170,000,000.00 |      6.25%         |   Good
2    |      $189,700,000.00 |      11.59%        |   Good
3    |      $191,000,000.00 |      0.69%         |   Good
Average Annual Growth Rate: 6.17%
Compound Annual Growth Rate: 6.08%
Standard Deviation: 5.452

                      Country Y
------------------------------------------------------------
Year |         GDP Value      |    Growth Rate     |   Ranking
------------------------------------------------------------
1    |      $567,800,000.00 |      0.00%         |   Poor
2    |      $732,100,000.00 |      28.94%        |   Exceptional
3    |      $890,101,100.00 |      21.58%        |   Good
Average Annual Growth Rate: 16.84%
Compound Annual Growth Rate: 16.17%
Standard Deviation: 15.040

                      Country Z
------------------------------------------------------------
Year |         GDP Value      |    Growth Rate     |   Ranking
------------------------------------------------------------
1    |      $452,200,000.00 |      10.29%        |   Good
2    |      $763,020,290.00 |      68.74%        |   Exceptional
3    |      $987,376,183.00 |      29.40%        |   Exceptional
Average Annual Growth Rate: 36.14%
Compound Annual Growth Rate: 34.04%
Standard Deviation: 29.799


The Country X's economy is most stable
```

Set 3

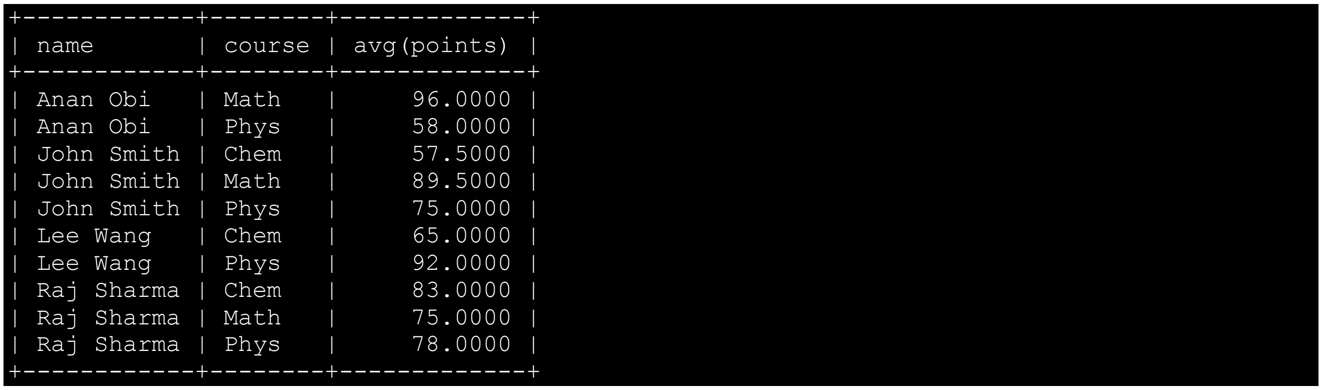
SQ 3.1

The following table shows the academic performance of students in a semester.

Student ID	Student Name	Phone	Course	Points
00012	John Smith	902-5556	Mathematics	86
00014	Raj Sharma	902-8596	Mathematics	75
00016	Anan Obi	902-8974	Mathematics	96
00015	Lee Wang	902-7845	Physics	92
00012	John Smith	902-5556	Physics	63
00016	Anan Obi	902-8974	Physics	58
00014	Raj Sharma	902-8596	Physics	78
00014	Raj Sharma	902-8596	Chemistry	83
00015	Lee Wang	902-7845	Chemistry	65
00012	John Smith	902-5556	Chemistry	95

1. There are functional dependencies in this table, so first, normalize the relation/table to the second normal form (2NF). Please show the resulting table(s) after the normalization.
2. Using the table(s) in 2NF, use a query to display the **average points** per course for each student, sorted by student names.

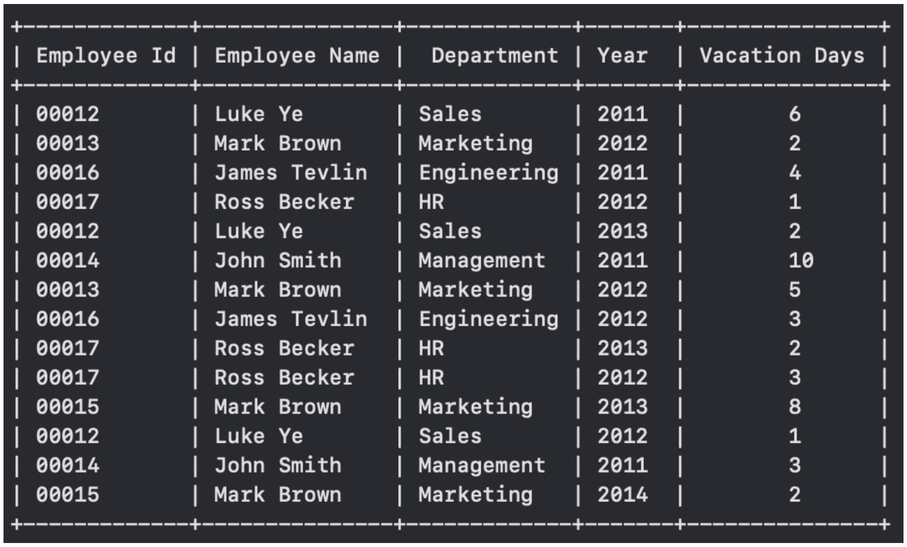
Correct output from the query



```
+------------+--------+-------------+
| name       | course | avg(points) |
+------------+--------+-------------+
| Anan Obi   | Math   |     96.0000 |
| Anan Obi   | Phys   |     58.0000 |
| John Smith | Chem   |     57.5000 |
| John Smith | Math   |     89.5000 |
| John Smith | Phys   |     75.0000 |
| Lee Wang   | Chem   |     65.0000 |
| Lee Wang   | Phys   |     92.0000 |
| Raj Sharma | Chem   |     83.0000 |
| Raj Sharma | Math   |     75.0000 |
| Raj Sharma | Phys   |     78.0000 |
+------------+--------+-------------+
```

SQ 3.2

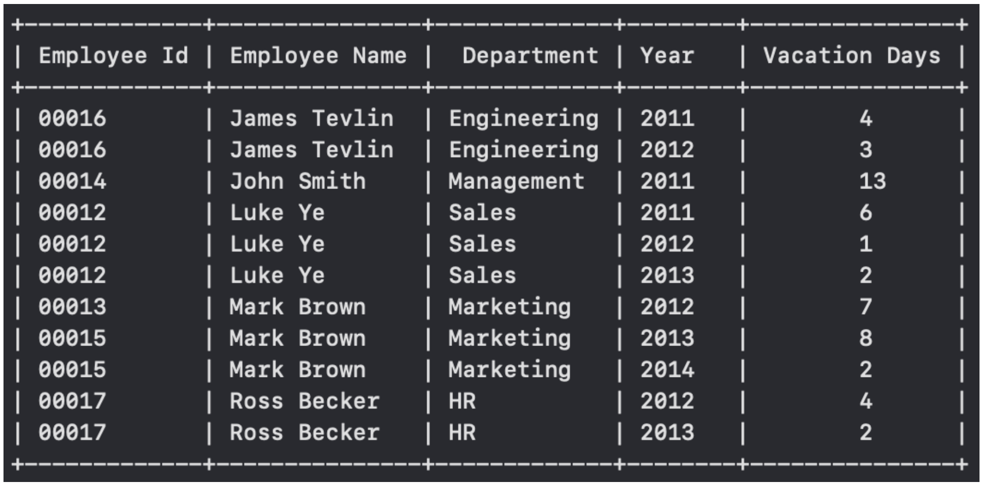
The following table shows vacation days taken by employees.



```
+-------------+---------------+-------------+-------+---------------+
| Employee Id | Employee Name |  Department | Year  | Vacation Days |
+-------------+---------------+-------------+-------+---------------+
| 00012       | Luke Ye       | Sales       | 2011  |       6       |
| 00013       | Mark Brown    | Marketing   | 2012  |       2       |
| 00016       | James Tevlin  | Engineering | 2011  |       4       |
| 00017       | Ross Becker   | HR          | 2012  |       1       |
| 00012       | Luke Ye       | Sales       | 2013  |       2       |
| 00014       | John Smith    | Management  | 2011  |       10      |
| 00013       | Mark Brown    | Marketing   | 2012  |       5       |
| 00016       | James Tevlin  | Engineering | 2012  |       3       |
| 00017       | Ross Becker   | HR          | 2013  |       2       |
| 00017       | Ross Becker   | HR          | 2012  |       3       |
| 00015       | Mark Brown    | Marketing   | 2013  |       8       |
| 00012       | Luke Ye       | Sales       | 2012  |       1       |
| 00014       | John Smith    | Management  | 2011  |       3       |
| 00015       | Mark Brown    | Marketing   | 2014  |       2       |
+-------------+---------------+-------------+-------+---------------+
```

1. There are functional dependencies in this table, so first, normalize the relation/table to the second normal form (2NF). Please show the resulting table(s) after the normalization.
2. Using the table(s) in 2NF, write a query to display the **total vacation days** per year for each employee, sorted by employee name and year.

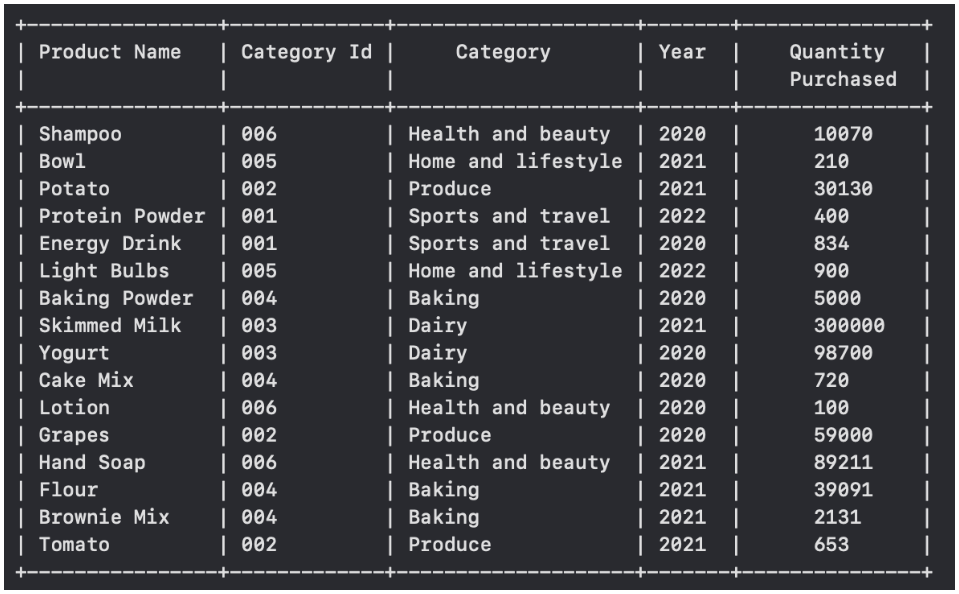
Correct output from the query



```
+-------------+---------------+-------------+--------+---------------+
| Employee Id | Employee Name |  Department | Year   | Vacation Days |
+-------------+---------------+-------------+--------+---------------+
| 00016       | James Tevlin  | Engineering | 2011   |       4       |
| 00016       | James Tevlin  | Engineering | 2012   |       3       |
| 00014       | John Smith    | Management  | 2011   |       13      |
| 00012       | Luke Ye       | Sales       | 2011   |       6       |
| 00012       | Luke Ye       | Sales       | 2012   |       1       |
| 00012       | Luke Ye       | Sales       | 2013   |       2       |
| 00013       | Mark Brown    | Marketing   | 2012   |       7       |
| 00015       | Mark Brown    | Marketing   | 2013   |       8       |
| 00015       | Mark Brown    | Marketing   | 2014   |       2       |
| 00017       | Ross Becker   | HR          | 2012   |       4       |
| 00017       | Ross Becker   | HR          | 2013   |       2       |
+-------------+---------------+-------------+--------+---------------+
```

SQ 3.3

Using **SQL** to perform the below functions on the following table showing a list of products sold.

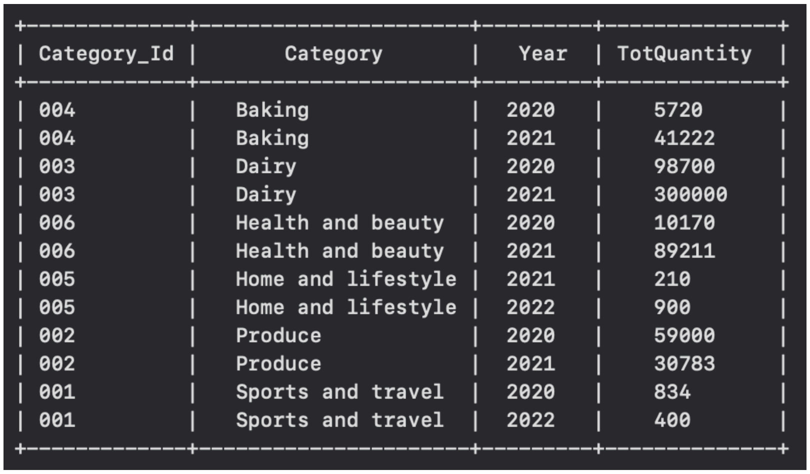


```
+----------------+-------------+--------------------+------+---------------+
| Product Name   | Category Id |      Category      | Year |   Quantity    |
|                |             |                    |      |   Purchased   |
+----------------+-------------+--------------------+------+---------------+
| Shampoo        | 006         | Health and beauty  | 2020 |     10070     |
| Bowl           | 005         | Home and lifestyle | 2021 |     210       |
| Potato         | 002         | Produce            | 2021 |     30130     |
| Protein Powder | 001         | Sports and travel  | 2022 |     400       |
| Energy Drink   | 001         | Sports and travel  | 2020 |     834       |
| Light Bulbs    | 005         | Home and lifestyle | 2022 |     900       |
| Baking Powder  | 004         | Baking             | 2020 |     5000      |
| Skimmed Milk   | 003         | Dairy              | 2021 |     300000    |
| Yogurt         | 003         | Dairy              | 2020 |     98700     |
| Cake Mix       | 004         | Baking             | 2020 |     720       |
| Lotion         | 006         | Health and beauty  | 2020 |     100       |
| Grapes         | 002         | Produce            | 2020 |     59000     |
| Hand Soap      | 006         | Health and beauty  | 2021 |     89211     |
| Flour          | 004         | Baking             | 2021 |     39091     |
| Brownie Mix    | 004         | Baking             | 2021 |     2131      |
| Tomato         | 002         | Produce            | 2021 |     653       |
+----------------+-------------+--------------------+------+---------------+
```

1. There are functional dependencies in this table, so first, normalize the relation/table to the second normal form (2NF). Please show the resulting table(s) after the normalization.
2. Using the table(s) in 2NF, write a query to display the **total quantity purchased** per year for each category, sorted by category and year.

(N.B. for this question we are just looking for the SQL table(s) and query)

Correct output from the query



```
+-------------+----------------------+---------+-------------+
| Category_Id |       Category       |  Year   | TotQuantity |
+-------------+----------------------+---------+-------------+
| 004         |  Baking              |  2020   |    5720     |
| 004         |  Baking              |  2021   |    41222    |
| 003         |  Dairy               |  2020   |    98700    |
| 003         |  Dairy               |  2021   |    300000   |
| 006         |  Health and beauty   |  2020   |    10170    |
| 006         |  Health and beauty   |  2021   |    89211    |
| 005         |  Home and lifestyle  |  2021   |    210      |
| 005         |  Home and lifestyle  |  2022   |    900      |
| 002         |  Produce             |  2020   |    59000    |
| 002         |  Produce             |  2021   |    30783    |
| 001         |  Sports and travel   |  2020   |    834      |
| 001         |  Sports and travel   |  2022   |    400      |
+-------------+----------------------+---------+-------------+
```