**CIS 106 – Problem Set 7 – IPO Charts**

For each problem prepare an IPO chart. Then write the code for each. Save the IPO within this document and upload it to your repository. After the code is complete upload the files (.py) to your repository. Paste the link to your repository into the assignment completion link in Blackboard.

1. Allow the user to enter a principal amount and interest rate repeatedly (need a loop to control the program execution). Compute the annual interest (principal x rate). Compute ending balance to be principal (beginning balance + interest). Display year, beginning balance and ending balance for each of the 5 years. Display the accumulated interest for the 5 years. Note: the new balance by year (this will be the principle for the following year. Format the output.

Example:

Enter principle amount: 10000.00

Enter interest rate: 0.10

Year Beginning Ending

Balance Balance

1. $10,000.00 $11,000.00
2. $11,000.00 $12,100.00
3. $12,100.00 $13,310.00
4. $13,310.00 $14,641.00
5. $14,641.00 $16,105.00

Total interest earned: $6,156.00

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESSES** | **OUTPUTS** |
| P = Principal Amount (USD) | InterestCumulative = 0  Display Year, Begin Balance (BB), and End Balance (EB) titles | Year, BB, EB titles |
| R = Interest Rate (decimal) | For range (1, 6):   * I = P \* R * EB = P + I * InterestCumulative = InterestCumulative + I * Display count (year), BB, EB * P = EB | For each count (year):   * Year * BB * EB |
|  | Display InterestCumulative | InterestCumulative |

1. Fibonacci sequence is a sequence of natural order. The sequence is: 1, 1, 2, 3, 5, 8 etc.

Use for loop compute and display first 20 numbers in the sequence. Hint: start with 1, 1.

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESSES** | **OUTPUTS** |
| a (first number) = 1 | Display a, b | a, b (restated input values) |
| b (second number) = 1 | for range (1, 21, 1) | c (20 times) |
|  | c = a + b |  |
|  | Display c |  |
|  | a = b, b = c |  |

1. Create a text file that contains the employees’ last name and salary. Read this data from the text file. Determine the bonus rate based on the chart below. Use that rate to compute bonus. For each line display the employees’ last name, salary, and bonus. After the loop display the sum of all bonuses paid out.

Salary Bonus Rate

100,000.00 and up 20%

50,000.00 15%

All other salaries 10%

Example file (create your own data with at least 5 lines:

Adams

50000.00

Baker

75000.00

Smith

45000.00

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESSES** | **OUTPUTS** |
| Text file containing:   * last names * salary | TotalBonus = 0  c = 0 (count) |  |
|  | get LastName  while LastName != “ ”  Get Salary  If Salary >= 100000:  BonusRate = 0.20  Else If Salary >= 50000:  BonusRate = 0.15  Else:  BonusRate = 0.10 | For each last name:   * last name * salary * bonus |
|  | Bonus = Salary \* BonusRate  TotalBonus = TotalBonus + Bonus  c = c + 1  Display last name, salary, bonus  get next last name |  |
|  | Avg. bonus = total bonus / c  Display c, total bonus, avg. bonus | c (final)  Total Bonuses  Average Bonus |

1. Create a text file with item, quantity, and price. Read through the file one line at a time. Compute the extended price (quantity x price). For each line display the item, quantity, price, and extended price. After the loop display the sum of all the extended prices, the count of the number of orders and the average order.

Example Data File:

Widget

10

50

Hammer

2

10

Saw

4

8

(Etc.)

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESSES** | **OUTPUTS** |
| Text file containing:   * Tool name * Tool quantity * Tool unit price | Sum Extended Price = 0  c = 0 (count) |  |
|  | get ToolName  while ToolName != “ ”  Get Tool quantity  Get Tool unit price  Extended Price = tool quantity \* tool price |  |
|  | c = c + 1  Sum Extended Price = Sum Extended Price + Tool Extended Price  Display tool name, tool quantity, tool unit price, tool extended price  get next Tool name | For each tool name:   * Tool name * Tool quantity * Tool unit price * Tool extended price |
|  | Display c, sum of extended prices, average extended price | c (final)  Sum of Extended Prices  Average Extended Price |

1. Create a text file with the students’ last name, district code (I or O) and number of credits taken. Compute the tuition owed (credits taken x cost per credit). Cost per credit for in district students (district code I) is 250.00. Out of the district students pay 500.00 per credit. For each line display student last name, credits taken, and tuition owed. After the loop display sum of all tuition owed and the number of students.

Example file:

Jones

I

12

Adams

I

10

Baker

O

12

Smith

O

16

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESSES** | **OUTPUTS** |
|  | TotalTuition = 0  C = 0 |  |
|  | Get first LastName |  |
| Text file containing:   * LastName * DistrictCode * Credits | While LastName != “ ”  Get DistrictCode, Credits  If DistrictCode = ‘I’:  CreditUnitCost = 250  Else:  CreditUnitCost = 500  Tuition = CreditUnitCost \* Credits  TotalTuition = TotalTuition + Tuition  C = C + 1  Display LastName, Credits, Tuition  Get next LastName | For each last name:   * LastName * Credits * Tuition |
|  | Display TotalTuition  Display C | TotalTuition  C |