#1

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1.Principal amount  2.Interest Rate | Get the principal amount from the user.  Get the interest rate from the user.  Loop through 5 years  Compute annual interest (principal \* rate).  Compute ending balance  (beginning balance + interest).  Update principal for the next year.  Calculate total accumulated interest over 5 years. | Formatted table with year, beginning balance, and ending balance  Total interest earned over 5 years |

#2

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1.No user input needed | Initialize the first two numbers of the Fibonacci sequence.  Use a for loop to generate the next 18 numbers (since the first two are already initialized).  Compute the next Fibonacci number.  Update values.  Continue looping until 20 numbers are generated. | Display the first 20 numbers of the Fibonacci sequence.  Each number is displayed in sequence. |

#3

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1.A text file containing employee last names and salaries. | Read each employee's last name and corresponding salary.  Determine the bonus rate based on the salary:   1. If salary >= 100,000.00 → bonus rate = 20% 2. f salary = 50,000.00 → bonus rate = 15%  |  | | --- | | 1. c. Otherwise → bonus rate = 10%   Compute the bonus using  Bonus= salary \* rate.  Display last name, salary, and calculated bonus.  Keep a running total of all bonuses.  After processing all employees, display the sum of all bonuses. |  |  | | --- | |  | | Employee's last name, salary, and computed bonus.  Sum of all bonuses paid out. |

#4

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1.A text file containing item names, quantities, and prices. | Read each line and extract the item name, quantity, and price.  Compute the extended price using: extendedPrice = quantity × price.  Display the item, quantity, price, and computed extended price.  Maintain a running total of all extended prices.   |  | | --- | | Keep track of the count of orders. |  |  | | --- | |  |   After processing all lines, compute the average order as: average Order = totalExtendedPrice / countOfOrders.  Display the total sum of extended prices, the number of orders, and the average order value. | Item name, quantity, price, and extended price for each item.  Sum of all extended prices.  Total count of orders  processed.  Average order value. |

#5

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 1.A text file containing student last name, district code (I or O), and number of credits taken. | Read each student's last name, district code, and number of credits taken.  Determine the cost per credit:   1. If district code, is I (in-district), cost per credit = 250.00 2. If district code is O (out-of-district), cost per credit = 500.00   Compute tuition owed using: tuition = creditsTaken × costPerCredit.  Display the student's last name, credits taken, and computed tuition.  Maintain a running total of all tuition amounts.  Keep track of the number of students processed.  After processing all students, display the total tuition owed and the total number of students. | Student's last name, credits taken, and tuition owed.  Sum of all tuition owed.  Total number of students. |