Assignments are deliberately kept at higher level of complexity to test both technical and logical reasoning skills. Please attempt whatever you can. Partial answers will be duly considered for evaluation.

# Using Any Programming Language of your choice

Write a program in any language of your choice (C, Java, SQL Stored Procedure, Unix Shell Programming, Python). Focus on getting the logic accurate rather than 100% technically correct syntax, as evaluation is more on your ability to logically design the construct.

The problem Statement

The program takes input a short abstract and outputs the following

* prints each word in reverse.
* prints Top 2 occurring words in sorted order. The words that have occurred most of the time.

Sample Input:

"This declaration represents a political commitment among declaration partners to advance a positive vision for the Internet in this era of a united europe"

Sample Output: Printing word in reverse

“Europe of century 21st of era this in Internet the for vision positive a advance to partners declaration among commitment political a represents declaration this”

Sample Output: Occurrence of words

a : 3 times

this : 2 times

# Assignment Using SQL

## ASSIGNMENT-01SQL

Please refer to two tables of Appendix A – Purchase History and Product Catalogue. Purchase History contains all purchases done for Grocery Store. Product Catalogue contains all product and its category.

Develop a SQL query that will find out two Products for each product category that are most popular in last 30 days. Popularity is based on maximum quantity sold in a particular category.

Sample Input:

Refer to Table in Appendix A

Sample Output:

|  |  |  |
| --- | --- | --- |
| Cat\_Id | Product\_Id | Trending |
| 1 | 100 | 1 |
| 1 | 200 | 2 |
| 2 | 300 | 1 |
| 2 | 301 | 2 |

## 

## ASSIGNMENT-02SQL

There is a Customer Table having a single column with list of customer id. There is a Voucher table having a single column with list of voucher ids.

Develop a query that will assign one voucher to one customer and vice versa. Two customers will not get same voucher. Two Voucher will not be assigned to a single customer.

Sample Input

|  |
| --- |
| Customer\_Id |
| Abhinash |
| Vipin |
| Mahesh |
| Bijoy |
| Bhabani |
| Ashutosh |

|  |
| --- |
| Voucher\_Id |
| ABXFH |
| SDFGH |
| ERTYY |
| PPLKM |

|  |  |
| --- | --- |
| Customer\_Key | Gift Voucher Key |
| Abhinash | ABXFH |
| Vipin | SDFGH |
| Mahesh | ERTYY |
| Bijoy | PPLKM |
| Bhabani | - |
| Ashutosh | - |

Sample Output

# Assignment Using Unix Shell Programming (Optional)

## ASSIGNMENT-01UNIX

There is a data file containing purchase history data. Write a script to find out list of unique dates on which there are sales.

Sample Input :.

***Product, Customer, Date, Category, amt***

*Nylon V Neck, Rakesh, 2020-01-02, Shirt,799.00*

*Nylon Y Neck, Ramesh, 2020-01-02, Shirt,799.00*

*Nylon Z Neck, Rajesh, 2020-01-02, Shirt,899.00*

*Nylon Z Neck, Rajesh, 2020-01-03, Shirt,699.00*

Sample Output

*2020-01-02*

*2020-01-03*

## ASSIGNMENT-02UNIX

There is a folder that contains everyday purchase history of a Grocery store. There is one file for each day and all files has same format. All files have a header.

Write a unix shell script to merge all files into one file. Merged files must have only 1 header.

Sample Input

*/home/grocery/purchase\_history\_02Jan21.txt*

*/home/grocery/purchase\_history\_03Jan21.txt*

*/home/grocery/purchase\_history\_31Jan21.txt*

Sample Output

*/home/grocery/purchase\_history\_JanAll.txt*

APPENDIX A

Purchase History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bills\_Id | Customer | Product\_Id | Sale\_Qty | Bill Amount | Bill\_Date |
| 78807159 | 44 | 100 | 1 | 16000 | 22-02-2020 |
| 82390697 | 44 | 100 | 2 | 5200 | 23-02-2020 |
| 94479024 | 44 | 100 | 1 | 4200 | 27-02-2020 |
| 94549714 | 44 | 100 | 1 | 21900 | 27-02-2020 |
| 95521191 | 44 | 200 | 1 | 20000 | 27-02-2020 |
| 142109733 | 44 | 200 | 1 | 8400 | 11-11-2019 |
| 158391727 | 44 | 300 | 1 | 24100.01 | 16-11-2019 |
| 168354993 | 44 | 300 | 1 | 24100.01 | 19-11-2019 |
| 193504854 | 44 | 300 | 1 | 7600 | 26-11-2019 |
| 195567022 | 44 | 300 | 1 | 2500 | 26-11-2019 |
| 216619949 | 44 | 301 | 1 | 16000 | 01-12-2019 |
| 228647858 | 44 | 401 | 1 | 28500.01 | 04-12-2019 |
| 244924788 | 66 | 402 | 1 | 18200 | 08-12-2019 |
| 244938525 | 66 | 403 | 1 | 16000 | 08-12-2019 |
| 244947344 | 66 | 404 | 2 | 12600 | 08-12-2019 |
| 245391938 | 66 | 405 | 1 | 4200 | 08-12-2019 |
| 245896327 | 66 | 406 | 1 | 12500 | 08-12-2019 |
| 270582683 | 66 | 407 | 1 | 5900 | 16-12-2019 |
| 271089616 | 66 | 405 | 1 | 22500 | 16-12-2019 |
| 393252504 | 66 | 301 | 1 | 26100 | 19-01-2020 |
| 395961027 | 66 | 401 | 1 | 12900 | 19-01-2020 |
| 407474852 | 77 | 402 | 1 | 4900 | 23-01-2020 |

Product Catalogue

|  |  |  |
| --- | --- | --- |
| Product\_Id | Dep\_Id | Cat\_Id |
| 100 | 100 | 1 |
| 100 | 100 | 1 |
| 100 | 100 | 1 |
| 100 | 100 | 1 |
| 200 | 200 | 1 |
| 200 | 200 | 1 |
| 300 | 300 | 2 |
| 300 | 300 | 2 |
| 300 | 300 | 2 |
| 301 | 301 | 2 |
| 401 | 401 | 3 |
| 403 | 403 | 3 |
| 404 | 404 | 3 |