### cs370 program 4 fall 2022

### Dr. Kyrianne Drakea is a brilliant scientist. Her company, Healing Vibes, sells healing oils and creams. Her business has grown rapidly. Every month, details of Healing Vibes inventory are gathered together into one large Inventory File from her 4 warehouses. Supplies are purchased from 3 Vendors.

### Dr. Drakea has asked you to write a program that:

### Sort the large input file (PR4F22-VibesInven.TXT) in Ascending Order on 3 key fields (WAREHOUSE, VENDERID AND PRODUCTID)

### Divide the now sorted large input file into 4 separate data files based on the Warehouse key field and 1 error file. (PR4F22-LAX1.TXT, PR4F22-SEA1.TXT, PR4F22-SLC1.TXT, PR4F22-SLC2.TXT and ERROR.TXT). Note - You are creating data files not a report file in this step.

### Merge the PR4F22-SLC1.TXT and the PR4F22-SLC2.TXT into 1 file UTAH-FILE.TXT.

### Open the UTAH-FILE.TXT file you created and use it as input to write a detailed report (Multi-Level Control Break). See Printer Spacing Chart

PR4F22-VibesInven.TXT INPUT: 128-character record on disk (UNSORTED COMBINED FILE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CC | FIELD | |  | |
| 1-4 | Warehouse ID | AN | LAX1, SEA1, SLC1, SLC2 | |
| 5 | Vender ID | A | M-Mad Hatter Oils, P-Pure Creams, C-Cheebs Herbs | |
| 6-8 | Product ID | AN | C01-C10, O01-O10 | |
| 9-128 | Product Data Array (maximum of 5 elements) | | Product Name - AN | X(13) |
| Product Size - A | X- Extra Large, L – Large,  M – Medium, S – Small,  A-Sample |
| Product Type - A | C – Cream, O – Oil |
| Number in stock - N | 9(4) |
| Purchase price - N | S999v99 |

SORTEDVIBESINVEN.TXT INPUT: 128-character record on disk (SORTED COMBINED FILE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CC | FIELD | | LAX1, SEA1, SLC1, SLC2 | |
| 1-4 | Warehouse ID | AN |  | |
| 5 | Vender ID | A | M-Mad Hatter Oils, P-Pure Creams, C-Cheebs Herbs | |
| 6-8 | Product ID | AN | C01-C10, O01-O10 | |
| 9-128 | Product Data Array (maximum of 5 elements) | | Product Name - AN | X(13) |
| Product Size - A | X- Extra Large, L – Large,  M – Medium, S – Small,  A-Sample |
| Product Type - A | C – Cream, O – Oil |
| Number in stock - N | 9(4) |
| Purchase price - N | S999v99 |

PR4F22-LAX1.TXT INPUT: 128-character record on disk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CC | FIELD | |  | |
| 1-4 | Warehouse ID | AN | LAX1 | |
| 5 | Vender ID | A | M-Mad Hatter Oils, P-Pure Creams, C-Cheebs Herbs | |
| 6-8 | Product ID | AN | C01-C10, O01-O10 | |
| 9-128 | Product Data Array (maximum of 5 elements) | | Product Name - AN | X(13) |
| Product Size - A | X- Extra Large, L – Large,  M – Medium, S – Small,  A-Sample |
| Product Type - A | C – Cream, O – Oil |
| Number in stock - N | 9(4) |
| Purchase price - N | S999v99 |

PR4F22-SEA1.TXT INPUT: 128-character record on disk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CC | FIELD | |  | |
| 1-4 | Warehouse ID | AN | SEA1 | |
| 5 | Vender ID | A | M-Mad Hatter Oils, P-Pure Creams, C-Cheebs Herbs | |
| 6-8 | Product ID | AN | C01-C10, O01-O10 | |
| 9-128 | Product Data Array (maximum of 5 elements) | | Product Name - AN | X(13) |
| Product Size - A | X- Extra Large, L – Large,  M – Medium, S – Small,  A-Sample |
| Product Type - A | C – Cream, O – Oil |
| Number in stock - N | 9(4) |
| Purchase price - N | S999v99 |

PR4F22-SLC1.TXT INPUT: 128-character record on disk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CC | FIELD | |  | |
| 1-4 | Warehouse ID | AN | SLC1 | |
| 5 | Vender ID | A | M-Mad Hatter Oils, P-Pure Creams, C-Cheebs Herbs | |
| 6-8 | Product ID | AN | C01-C10, O01-O10 | |
| 9-128 | Product Data Array (maximum of 5 elements) | | Product Name - AN | X(13) |
| Product Size - A | X- Extra Large, L – Large,  M – Medium, S – Small,  A-Sample |
| Product Type - A | C – Cream, O – Oil |
| Number in stock - N | 9(4) |
| Purchase price - N | S999v99 |

PR4F22-SLC2.TXT INPUT: 128-character record on disk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CC | FIELD | |  | |
| 1-4 | Warehouse ID | AN | SLC2 | |
| 5 | Vender ID | A | M-Mad Hatter Oils, P-Pure Creams, C-Cheebs Herbs | |
| 6-8 | Product ID | AN | C01-C10, O01-O10 | |
| 9-128 | Product Data Array (maximum of 5 elements) | | Product Name - AN | X(13) |
| Product Size - A | X- Extra Large, L – Large,  M – Medium, S – Small,  A-Sample |
| Product Type - A | C – Cream, O – Oil |
| Number in stock - N | 9(4) |
| Purchase price - N | S999v99 |

ERROR.TXT INPUT: 128-character record on disk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CC | FIELD | |  | |
| 1-4 | Warehouse ID | AN |  | |
| 5 | Vender ID | A | M-Mad Hatter Oils, P-Pure Creams, C-Cheebs Herbs | |
| 6-8 | Product ID | AN | C01-C10, O01-O10 | |
| 9-128 | Product Data Array (maximum of 5 elements) | | Product Name - AN | X(13) |
| Product Size - A | X- Extra Large, L – Large,  M – Medium, S – Small,  A-Sample |
| Product Type - A | C – Cream, O – Oil |
| Number in stock - N | 9(4) |
| Purchase price - N | S999v99 |

UTAH-FILE.TXT: 128-character record on disk (USED TO WRITE THE REPORT)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CC | FIELD | |  | |
| 1-4 | Warehouse ID | AN | SLC1, SLC2 | |
| 5 | Vender ID | A | M-Mad Hatter Oils, P-Pure Creams, C-Cheebs Herbs | |
| 6-8 | Product ID | AN | C01-C10, O01-O10 | |
| 9-128 | Product Data Array (maximum of 5 elements) | | Product Name - AN | X(13) |
| Product Size - A | X- Extra Large, L – Large,  M – Medium, S – Small,  A-Sample |
| Product Type - A | C – Cream, O – Oil |
| Number in stock - N | 9(4) |
| Purchase price - N | S999v99 |

INSTRUCTIONS

1. Sort the large input file (**PR4F22-VibesInven.TXT**) in Ascending Order on 3 key fields (WAREHOUSE, VENDERID AND PRODUCTID)
   1. **The keys for sorting are:**
      1. **Warehouse ID – major**
      2. **Vendor ID – intermediate**
      3. **Product ID – minor**
2. Divide the now sorted large input file into **5 separate files** based on the **Warehouse ID key field** and **REMOVE** any records (the entire record) if the warehouse ID is invalid. **\***Note - You are creating data files not a report file in this step so no editing symbols.
   1. Open the sorted file as input
   2. Look at the Warehouse ID
   3. If the Warehouse ID is valid write the entire record out to the **appropriate** file
   4. If Warehouse ID is invalid, write the entire record out to an ERROR file and do not include it in the new files.
   5. Keep a count of how many invalid records were written out and
   6. display a nice message on the screen at the end of the program informing the user of the number of invalid records.
   7. The File Names should be:
      1. **PR4F22-LAX1.TXT**
      2. **PR4F22-SEA1.TXT**
      3. **PR4F22-SLC1.TXT**
      4. **PR4F22-SLC2.TXT**
      5. **ERROR.TXT**
3. Merge the **PR4F22-SLC1.TXT** and the **PR4F22-SLC2.TXT** into 1 file called **UTAH-FILE.TXT** on
   1. **The keys for merging are:**
      1. **Warehouse ID – major**
      2. **Vendor ID – intermediate**
      3. **Product ID - minor**
4. Open the **UTAH-FILE.TXT** file you created as INPUT and use it to write a detailed report. The report is a multi-level control break with arrays and a table. See Printer Spacing Chart
   1. **The keys for breaking are:**
      1. **Warehouse ID – major**
      2. **Vendor ID – intermediate**
      3. **Product ID - minor**

REPORT INFO:

1. The Warehouse-ID is to be expanded.
   1. Put UTAH WAREHOUSE 1 in the Warehouse Header Field and the Warehouse Total Line Name when the warehouse id is SLC1
   2. Put UTAH WAREHOUSE 2 in the Warehouse Header Field and the Warehouse Total Line Name Field when the warehouse id is SLC2.
2. The Vendor ID is to be expanded via a Table lookup.
   1. Hard code the vendor information in working storage in a table.
   2. Search the Vendor Table.
   3. When found put the full vendor’s name in the Vendor Header Name Field **and** Vendor Total Name Field
   4. If the Vendor is not found in the table use concatenation or reference modification move to put “INVALID” and the invalid ID in the Vendor Header Name Field and the Vendor Total Name Field.
3. Only Print the Product name once in the detail line and print it in the Product Total Line.
4. The Product Sizing is to be expanded via an EVALUATE statement. If the Product Size is not found use concatenation or reference modification move to enter “BAD” and the invalid code in place of the sizing information.
5. The Product Type should be expanded and validated.
   1. If it is not found or is incorrect, use concatenation or reference modification move to enter BAD and the invalid code in place of the info.
6. The incoming record contains an array of up to **five** **elements**; blanks appear in the extra fields when an incoming record has fewer than five elements (validation will be necessary).
7. Use a Multi-Level Control Break to print the total cost spent for each product, for each vendor, for each warehouse and then a grand total.

### Select and FD statements:

### 1 input unsorted file – PR4F22-VIBESINVEN.TXT

### 1 output sorted file – will be reopened as input to create the 4 individual warehouse files and 1 error file.

### 1 SD temp- file that the computer uses for the sorting/merge process - strictly used by the computer for sorting

### 1 output error file

### 2 output files that won’t be used again in this program - PR4F22-LAX1.TXT, PR4F22-SEA1.TXT

### 2 output files that will be merged into 1 file - PR4F22-SLC1.TXT, PR4F22-SLC2.TXT

### 1 merged file that will be used as input to print the report – UTAH-FILE.TXT

### 1 output report file

### CALCULATIONS:

### In Stock X Price = Total Cost

### Accumulate Product Total Cost

### Accumulate Vendor Total Cost

### Accumulate Warehouse Total Cost

### Accumulated Grand Total Cost

### It is crucial that you sort the main input file first on the 3 keys in Ascending Order.

### Use the sorted file to create the PR4F22-LAX1.TXT, PR4F22-SEA1.TXT, PR4F22-SLC1.TXT, PR4F22-SLC2.TXT and ERROR files.

### Merge PR4F22-SLC1.TXT AND PR4F22-SLC2.TXT INTO UTAH-FILE.TXT

### Reopen UTAH-FILE.TXT as input and use it as input to write the report file.