

---

**Algorithm 1** Main Program Flow

---

```
1: Initialize 'sales_data' File if not exists
2: Create File and Write Headers if necessary
3: while Main Menu Loop do
4:     Display Main Menu
5:     switch userInput do
6:         case "1": Add Sales Data do
7:             Prompt for SalesPerson Details
8:             Write SalesPerson to 'sales_data'
9:         case "2": Exit do
10:            End Program
11:     Print Sales Data
12: end while
```

---

---

**Algorithm 2** PriorityQueue Class

---

```
1: function ENQUEUE(value)
2:     Call BinaryInsertion
3: end function
4: function DEQUEUE
5:     Remove and Return Last Element
6: end function
7: function BINARYSEARCH(value)
8:     Initialize Search Range
9:     while Searching do
10:        Calculate Midpoint
11:        Compare Priority Values
12:        Adjust Search Range
13:     end while
14:     return Position and Value
15: end function
```

---

---

**Algorithm 3** ColumnWriter Class

---

```
1: function ADD(value)
2:   Ensure Enough Columns Exist
3:   Add Value to Selected Column
4:   Update Column Widths
5:   Rotate Column Selector
6: end function
7: function WRITE
8:   Initialize StringBuilder
9:   for all Rows do
10:     for all Columns do
11:       Append Column Value and Spacing
12:     end for
13:   Append Line Break
14:   end for
15:   Print Content
16: end function
```

---

---

**Algorithm 4** Print Sales Data Procedure

---

```
1: procedure PRINTSALESDATA(path)
2:   Initialize salesTiers array
3:   Read all lines from 'sales_data' file
4:   Split first line to get headers
5:   Initialize ColumnWriter with headers length
6:   for each header in headers do
7:     Add header to ColumnWriter
8:   end for
9:   Initialize PriorityQueue for SalesPerson objects
10:  for each line in lines (excluding first line) do
11:    Split line to get values
12:    Enqueue SalesPerson object to PriorityQueue
13:  end for
14:  if PriorityQueue is empty then
15:    return
16:  end if
17:  Use BinarySearch to find boundary indices of sales tiers
18:  Initialize salesTiersIndices array
19:  for each salesTier in salesTiers do
20:    Find index for salesTier in PriorityQueue
21:    Store index in salesTiersIndices
22:  end for
23:  Initialize index variable
24:  while PriorityQueue has elements do
25:    Dequeue SalesPerson from PriorityQueue
26:    Add SalesPerson details to ColumnWriter
27:    if Current index matches any in salesTiersIndices or last element
then
28:      Determine tier level and range
29:      Add tier information to ColumnWriter
30:      Add empty columns and separator
31:    end if
32:    Increment index
33:  end while
34:  Write all column data using ColumnWriter
35: end procedure
```

---