Chapter 5
An Introduction to Data Structure

Objectives
- Method for defining constants and variables
- Storage and access requirements of arrays and strings
- Two-dimensional arrays
- Different techniques for accessing stored information
- Difference between static and dynamic storage allocation
- Operation of linked-lists and binary trees
- Operation of stacks and queues

Introduction
- Data comes in many forms, such as bits, bytes, words, or longwords
- BCD is another data type
- Also group large chunks of data together into structures, such as arrays, matrices, or linked-lists

Defining Constants with EQU
<label> EQU <value>

Defining Variables with DC and DS
<label> DC.<size> <value>
<label> DS.<size> <number>

Accessing One-Dimensional Arrays
- In high-level language such as C, we use statements:
  card = cards[i];
- In 68000 assembly language, we can use address register indirect with index addressing
Accessing Two-Dimensional Arrays

- In high-level language such as C, we use statements:
  \[ \text{val} = \text{magicsq}[\text{row}][\text{col}]; \]
- In 68000 assembly language, we can use address register indirect with index addressing

Character Strings

- A character string is a collection of ASCII character codes terminated by the byte value 00

Static Versus Dynamic Storage Allocation

- Static storage allocation: predefined, fixed lengths that do not change during execution of a program
- Dynamic storage allocation: Space required for data storage is assigned as requested, not in advance

Linked-Lists

- A linked-list is a collection of data elements called nodes that is created dynamically
- Inside of one node, the first item is for data, the second item is called a pointer

Binary Trees

- A binary tree is a specific data structure composed of nodes containing a data field and two link fields

Stacks and Queues

- Stack – is an area in memory reserved for reading and writing special data items such as return addresses and register values – LIFO (last in, first out)
- Queue – also memory-based structures, but its operation is functionally different from that of a stack – FIFO (first in, first out)
<table>
<thead>
<tr>
<th>Troubleshooting Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Know your data size</td>
</tr>
<tr>
<td>- Be familiar with various addressing modes</td>
</tr>
<tr>
<td>- Be familiar with signed and unsigned data ranges</td>
</tr>
<tr>
<td>- Structure your data to avoid address errors</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Methods for defining constants and variables, data structures that contain a single item</td>
</tr>
<tr>
<td>- Covered different data structures</td>
</tr>
</tbody>
</table>