Blank Page

0.1 Basic Data Structures and Programming: Lists and Trees

1. Program: 5-24, 5025 page 182, 193. of OP. Write a program to find and print all divisors of an input number. Find the number below 100 that has the largest number of divisors. Find the three numbers less than 100 which are perfect. And so on.

   Reading: OP–5: "The if and case Statements," especially page 177: "Simplifying boolean Evaluation." OP–2.2, Core Functions from SYSTEM. The Debugger Tutorials from OP–1-6, OP–2-5, and OP–6-5.

   Classwork: Control structures, program correctness, review of Pascal as required.

2. Program: The *pig-latin* program. Write a program that takes an input text and translates it into pig-latin according to two rules,

   (a) If the word begins with a consonant, move it to the end of the word before applying the next rule.

   (b) Add the suffix "ay" to the word.

   This is problem 7-18, page 267 in OP, where it is presented as:

   > Ifay ouyay ancay eadray histay, ouyay ancay robablypay igurefay outay hatway hetay roblempay isay.

   Goals: Reading text from input. Handling individual characters. Parsing words from a text.

   Reading: OP–7, Text Processing. OP–6: "The repeat and while Loops", especially OP–6-2 Loop Specification.

   Class work: Boolean algebra and loop specification. Examples, splitting an array. Comparing strings for order.

3. The *letter frequency* program. Write a program that counts the number of appearences of each letter a through z in an input text. Print out a bar chart showing the distribution. Upper and lower case letters are not distinguished, "A" is counted the same as "a", spaces and punctuation is ignored.

Use the program to break a code.

Reading: OP–9: “The array type”, especially OP–9-2, The Basic Loops.

Class work: Computer memory, arrays and records.

4. Program: The *unique words* program. Maintain a linked list of words in an input text where each word in the text appears only once on the list, with a integer count variable indicating how many times this word has been seen in the text. Add new words to the front of the list. At the EOF, write this list and the associated word counts.

   Goals: Construction of, traversal of and appending to a linked list.

   Reading: OP–11, The *record* Type, and OP–16, The *pointer* Type.

   Classwork: Computer hardward: memory, arrays, records and pointers. The list data structure.

5. Program: Adapt previous program so that new words are added to the end of the list. Rewrite to make the list an object with operations.

   Goal: Introducing Object Oriented Programming and Units.

   Reading: OP–13: “Units”. OP–14: “Objects”.

6. Program: Insert words into a search tree. Searching and Printing the tree by recursion.

   Goal: Introduction of tree data structure and Recursion.

   Reading: OP–17: “Recursive Programing.”

0.2 Basic Algorithms: Sorting and Searching

- Introduction, Big Oh notation.

  Reading: Algorithms 1–6.

- Sorting.

  Reading: Algorithms 7–13.

- Searching

  Reading: Algorithms 14–18.

0.3 And so on

Additional items:

- Program: Cards. AKA 13–10. Shuffling a hand of cards, using previos program and input the name of each card.

  What is the probability of winning this game:

  - The deck is shuffled and the 52 cards are placed in 13 piles of 4 cards each. The piles are arranged in a circle, with one pile in the center of the circle, so that each pile can be indicated by a position of a clock. That is, there is a pile at one o'clock, two o'clock, and so on.
  - Cards are turned over one by one, from the top of each pile, the value of the card just turned over telling you which pile to go to next. If the card just turned over was an Ace, turn over next the top card from the pile at one o'clock, if the card was a two, go next to the pile at two o'clock, and so on. The King indicates the center pile.
  - Begin the game by turning over the top card from the center file.
  - Eventually this game will end: you will be asked to turn over a card from a pile for which all four cards have already been turned over. If this happens at the last card, you win, else you loose.

  Question: what is the probablity of winning this game?