Supplementary Problems

ARRAYS

4.21 Consider the linear arrays XXX(−10:10), YYY(1935:1985), ZZZ(35). (*a*) Find the number of elements in each array. (*b*) Suppose *Base*(YYY) = 400 and $w = 4$ words per memory cell for YYY. Find the address of YYY[1942], YYY[1977] and YYY[1988].

4.22 Consider the following multidimensional arrays:

X(−5:5, 3:33) Y(3:10, 1:15, 10:20)

(*a*) Find the length of each dimension and the number of elements in X and Y.

(*b*) Suppose *Base*(Y) = 400 and there are $w = 4$ words per memory location. Find the effective indices E_1, E_2, E_3 and the address of Y[5, 10, 15] assuming (i) Y is stored in row-major order and (ii) Y is stored in column-major order.

4.23 An array A contains 25 positive integers. Write a module which

(*a*) Finds all pairs of elements whose sum is 25

(*b*) Finds the number EVNUM of elements of A which are even, and the number ODNUM of elements of A which are odd

4.24 Suppose A is a linear array with n numeric values. Write a procedure

MEAN(A, N, AVE)

which finds the average AVE of the values in A. The *arithmetic mean* or *average* $\bar{x}$ of the values $x_1, x_2, \ldots, x_n$ is defined by

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

4.25 Each student in a class of 30 students takes 6 tests in which scores range between 0 and 100. Suppose the test scores are stored in a 30×6 array TEST. Write a module which

(*a*) Finds the average grade for each test

(*b*) Finds the final grade for each student where the final grade is the average of the student's five highest test scores

(*c*) Finds the number NUM of students who have failed, i.e., whose final grade is less than 60

(*d*) Finds the average of the final grades

POINTER ARRAYS; RECORD STRUCTURES

4.26 Consider the data in Fig. 4-26(*c*). (*a*) Write a procedure which prints the list of clients belonging to LAWYER[K]. (*b*) Assuming CLIENT has space for 400 elements, define an array FREE such that FREE[K] contains the number of empty cells following the list of clients belonging to LAWYER[K].

4.27 The following is a list of entries, with level numbers, in a file of employee records:

1 Employee(200), 2 SSN(Social Security Number), 2 Name, 3 Last, 3 First, 3 MI (Middle Initial), 2 Address, 3 Street, 3 Area, 4 City, 4 State, 4 ZIP, 2 Age, 2 Salary, 2 Dependents

(a) Draw the corresponding hierarchical structure.

(b) Which of the items are elementary items?

(c) Describe a record structure—for example, a PL/1 structure or a Pascal record—to store the data.

4.28 Consider the data structure in Fig. 4-27. Write a procedure to carry out each of the following:

(a) Finding the number of new Oldsmobiles selling for under $10 000.

(b) Finding the number of new automobiles selling for under $10 000.

(c) Finding the number of automobiles selling for under $10 000.

(d) Listing all automobiles selling for under $10 000.

(*Note*: Parts (c) and (d) require only the arrays AUTO and PRICE together with the number of automobiles.)

4.29 A class of student records is organized as follows:

1 Student(35), 2 Name, 3 Last, 3 First, 3 MI (Middle Initial), 2 Major
2 Test(4), 2 Final, 2 Grade

(a) How many elementary items are there?

(b) Describe a record structure—for example, a PL/1 structure or a Pascal record, to store the data.

(c) Describe the output of each of the following Write statements: (i) Write: Final[15], (ii) Write: Name[15] and (iii) Write: Test[4].

4.30 Consider the data structure in Prob. 4.18. Write a procedure which

(a) Finds the average of the sophomore GPA scores

(b) Finds the number of biology majors

(c) Finds the number of CUM scores exceeding K

Programming Problems

ARRAYS

Assume that the data in Table 4-1 are stored in linear arrays SSN, LAST, GIVEN, CUM and YEAR (with space for 25 students) and that a variable NUM is defined which contains the actual number of students.

4.31 Write a program for each of the following:

(a) Listing all students whose CUM is K or higher. (Test the program using K = 3.00.)

(b) Listing all students in year L. (Test the program using L = 2, or sophomore.)

4.32 Translate the linear search algorithm into a subprogram LINEAR(ARRAY, LB, UB, ITEM, LOC) which either finds the location LOC where ITEM appears in ARRAY or returns LOC = 0.

4.33 Translate the binary search and insertion algorithm into a subprogram BINARY(ARRAY, LB, UB, ITEM, LOC) which finds either the location LOC where ITEM appears in ARRAY or the location LOC where ITEM should be inserted into ARRAY.

Table 4-1

Social Security Number	Last Name	Given Name	CUM	Year
211-58-1329	Adams	Bruce	2.55	2
169-38-4248	Bailey	Irene L.	3.25	4
166-48-5842	Cheng	Kim	3.40	1
187-52-4076	Davis	John C.	2.85	2
126-63-6382	Edwards	Steven	1.75	3
135-58-9565	Fox	Kenneth	2.80	2
172-48-1849	Green	Gerald S.	2.35	2
192-60-3157	Hopkins	Gary	2.70	2
160-60-1826	Klein	Deborah M.	3.05	1
166-52-4147	Lee	John	2.60	3
186-58-0430	Murphy	William	2.30	2
187-58-1123	Newman	Ronald P.	3.90	4
174-58-0732	Osborn	Paul	2.05	3
183-52-3865	Parker	David	1.55	2
135-48-1397	Rogers	Mary J.	1.85	1
182-52-6712	Schwab	Joanna	2.95	2
184-48-8539	Thompson	David E.	3.15	3
187-48-2377	White	Adam	2.50	2

4.34 Write a program which reads the social security number SOC of a student and uses LINEAR to find and print the student's record. Test the program using (*a*) 174-58-0732, (*b*) 172-55-5554 and (*c*) 126-63-6382.

4.35 Write a program which reads the (last) NAME of a student and uses BINARY to find and print the student's record. Test the program using (*a*) Rogers, (*b*) Johnson and (*c*) Bailey.

4.36 Write a program which reads the record of a student

SSNST, LASTST, GVNST, CUMST, YEARST

and uses BINARY to insert the record into the list. Test the program using:

(*a*) 168-48-2255, Quinn, Michael, 2.15, 3

(*b*) 177-58-0772, Jones, Amy, 2.75, 2

4.37 Write a program which reads the (last) NAME of a student and uses BINARY to delete the student's record from the list. Test the program using (*a*) Parker and (*b*) Fox.

4.38 Write a program for each of the following:

(*a*) Using the array SSN to define arrays NUMBER and PTR such that NUMBER is a sorted array of the elements in SSN and PTR[K] contains the location of NUMBER[K] in SSN.

(*b*) Reading the social security number SOC of a student and using BINAR and the array NUMBER to find and print the student's record. Test the program using (i) 174-58-0732, (ii) 172-55-5554 and (iii) 126-63-6382. (Compare with Prob. 4.34.)

Assume the data in Table 4-3 are stored in a single linear array CLASS (with space for 30 names), assume that there are 3 empty cells between the sections, and that there are linear arrays NUMB, PTR and FREE defined so that NUMB[K] contains the number of elements in Section K, PTR[K] gives the location in CLASS of the first name in Section K, and FREE[K] gives the number of empty cells in CLASS following Section K.

Table 4-3

Section 1	Section 2	Section 3	Section 4
Brown	Abrams	Allen	Burns
Davis	Collins	Conroy	Cohen
Jones	Forman	Damario	Evans
Samuels	Hughes	Harris	Gilbert
	Klein	Rich	Harlan
	Lee	Sweeney	Lopez
	Moore		Meth
	Quinn		Ryan
	Rosen		Williams
	Scott		
	Taylor		
	Weaver		

4.39 Write a program which reads an integer K and prints the names in Section K. Test the program using (*a*) K = 2 and (*b*) K = 3.

4.40 Write a program which reads the NAME of a student and finds and prints the location and section number of the student. Test the program using (*a*) Harris, (*b*) Rivers and (*c*) Lopez.

4.41 Write a program which prints the names in columns as they appear in Table 4-3.

4.42 Write a program which reads the NAME and section number SECN of a student and inserts the name into CLASS. Test the program using (*a*) Eden, 3; (*b*) Novak, 4; (*c*) Parker, 2; (*d*) Vaught, 1; (*e*) Bennett, 3. (The program should handle OVERFLOW.)

4.43 Write a program which reads the NAME of a student and deletes the student from CLASS. Test the program using (*a*) Klein, (*b*) Daniels, (*c*) Meth and (*d*) Harris.

MISCELLANEOUS

4.44 Suppose A and B are n-element vector arrays in memory and X and Y are scalars. Write a program which finds (*a*) XA + YB and (*b*) A · B. Test the program using A = (16, −6, 7), B = (4, 2, −3), X = 2 and Y = 3.

4.45 Translate the matrix multiplication algorithm, Algorithm 4.7, into a subprogram

MATMUL(A, B, C, M, P, N)

which finds the product C of an $m \times p$ matrix A and a $p \times n$ matrix B. Test the program using

$$A = \begin{pmatrix} 4 & -3 & 5 \\ 6 & 1 & -2 \end{pmatrix} \qquad B = \begin{pmatrix} 2 & 3 & -7 & -3 \\ 5 & -1 & 6 & 2 \\ 0 & 3 & -2 & 1 \end{pmatrix}$$