Exercise-29

For each exercise given below you can also download the corresponding text files for testing your program:

29.1. Systems planning involve estimating development time and projection of costs associated with the completion of a particular task. An example might be planning construction of a house. A list of tasks and a time schedule might be:

Sequence of Events	Task Description	Time in Days
1	Laying of foundation	15
2	Plumbing installment	12
3	Frame and roof	20
4	Electrical work	7
5	Plastering	4
6	Carpentry	7
7	Landscaping	3

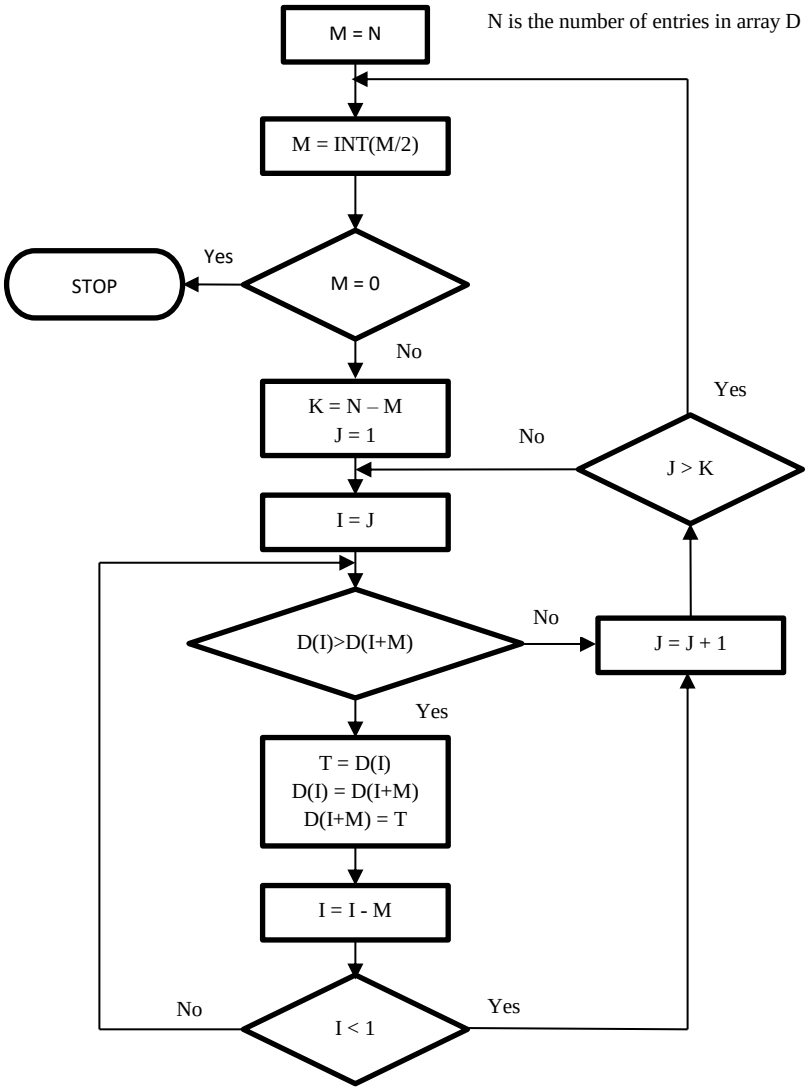
Write a program to read the input file and produce a bar graph like the following one to represent the duration of each task:

```
Tasks
1   ***************
2                  ************
3                              ********************
4                                                  *******
5                                                         ****
6                                                             *******
7                                                                    ***
    ---------------------------------------------------------------------------
    000000000111111111122222222223333333333444444444455555555556666666666
    123456789012345678901234567890123456789012345678901234567890123456789012345678

                              Time Duration Segments
```

29.2. Two sort methods are very popular to sort the numbers in the array, namely, bubble sort and the mini/max sort. These two sorting techniques are not the most efficient methods, however. An extremely efficient sorting method is the Shell-Metzner sort displayed here in flowchart form. In an article entitled “A Comparison of Sorts” in *Creating Computing*, Volume 2, John Grillo compare the three methods and determined that to sort 100,000 numbers would take 7.1 days, 3.8 days and 15 minutes for the bubble, mini/max, and Shell-Metzner sort, respectively. To sort 10,000,000 numbers would take 93 years, 50 years, and

2.5 days respectively! Write a program to sort N numbers read from the file using the following flowchart.



29.3. Write a program to convert decimal numbers to Roman numerals, employing usual rules required in Roman numerals. Recall that the basic correspondence between roman and decimal numerals is as follows:

Roman	**Decimal (Hindu-Arabic)**
M	1000
D	500
C	100
L	50
X	10
V	5
I	1

Write a program to read an input file and display the both the Decimal and the corresponding Roman number alongside. Make sure you get the following answers:

Decimal	**Expected Roman**	**Decimal**	**Expected Roman**
3	III	49	XLIX
4	IV	50	L
5	V	58	LVVIII
7	VII	75	LXXV
9	IX	90	XC
10	X	200	CC
12	XII	350	CCCL
24	XXIV	400	CD
29	XXIX	500	D
38	XXXVIII	900	CM
40	XL	2900	MMCM
44	XLIV		

29.4. Read word and write it in reverse order, then read and reverse a sentence one word at a time.

29.5. A very simple method to encode a message is to substitute each letter of the message with another according to some scheme such as:

Letter	A	B	C	D	E	…	Y	Z
Substitute	Z	Y	X	W	V	…	B	A

Write a program to read a file, encode and display in the output.

All the Best

•••

Exercise-30

For each exercise given below you can also download the corresponding text files for testing your program:

30.1. Loan an array of size 8 with any digits from 1 to 9, then write the code to store these eight digits as a single number in memory location KNUMB. Print the output KNUMB * 2 for verification.
For example, if A(1) = 3, A(2)=5, A(3)=2 then KNUMB=352. The output should be 704.

30.2. Read a three-digit number, N, and store each individual digit of N in three consecutive array locations. For example, if N = 193, then J(3) = 1, J(2) = 9, J(1) = 3.

Then print the array values for verification.

30.3. Write a program to compute n! = n(n-1) * (n-2) * … * 2 * 1 for values of n that are so large that the result cannot fit in a single integer memory location. For example, 25! = 15,511,210,043,330,985,984,000,000. One way to solve the problem is to use an array to store the answer (and partial results), using one array memory location per digit. Thus 12! = 479,001,600 would be stored as follows:

J(9)	J(8)	J(7)	J(6)	J(5)	J(4)	J(3)	J(2)	J(1)
4	7	9	0	0	1	6	0	0

To obtain 13!, multiply each memory location by 13 (taking care to move the carries) to obtain:

J(10)	J(9)	J(8)	J(7)	J(6)	J(5)	J(4)	J(3)	J(2)	J(1)
6	2	2	7	0	2	0	8	0	0

30.4. Two integer numbers containing a maximum 80 digits are contained on two lines of the given input file. Write a program to compute the sum of these numbers.

30.5. Each record in a file consists of six data items, a student name followed by five test scores, for example:

DOE	10	20	30	40	50
SLY	10	10	70	60	40

There are at most 100 records. Read the data into a two-dimensional table. For example, A(3,I), with I ranging from 1 to 5, represents the third student's test scores. Compute the average score for each student and the average score on each test. The output should have the following format:

Name	Test1	Test2	Test3	Test4	Test5	Average
DOE	10	20	30	40	50	30.0
SLY	11	10	71	60	40	38.0
Average/Test	10.5	15.0	50.5	50.0	45.0	

All the Best

•••

Exercise-28

For each exercise given below you can also download the corresponding text files for testing your program:

28.1. Each employee working at Slimpower, Inc. is paid at a regular hourly rate of $5 for the first 40 hours. The overtime rate is 1.5 times the regular rate. The number of hours worked by each employee is to be read from the input file.

Write a program to compute each employee's pay and produce an output similar to the following:

Hours	Rate	OT Hours	Pay
10.00	5.0	0	50.00
50.00	5.0	10	275.00
.	.	.	.
.	.	.	.
.	.	.	.

28.2. Computerized checkbook: The first record of an input file contains the balance from the previous month. Each succeeding record contains two entries: a check number and a dollar amount. This dollar amount can be a positive number (deposit) or a negative number (withdrawal). Write a program to produce a checkbook report similar to the following:

Input

3250	
112	-100
123	-250
127	50
128	74

Output

Check No.	Withdrawals	Deposits	Balance
			$3250.00
112	100.00		3150.00
123	250.00		2900.00
127		50.00	2950.00
128		74.00	3024.00
		New Balance	$3024.00

28.3. A data file consists of records, each containing the following information concerning items produced at the XYZ manufacturing plant: a depart number, an item number, a quantity, and a cost per item. Assume the file has been sorted into order by ascending department number. Write a program to produce a summary report as follows:

Department	Item	Quantity	Cost/Item	Value	Totals
15	1389	4	3.20	12.80	
15	3821	2	7.00	14.00	
					26.80
16	0122	8	2.50	20.00	
					20.00
19	1244	100	0.03	3.00	
19	1245	20	4.00	80.00	
19	2469	4	16.00	64.00	
					147.00
				Grand Total	193.80

Could you alter the program to write each subtotal on the same line as the last entry for each department?

28.4. You own a bookstore that sells both paperback and hardback books. For every book you sell, you have a record with two numbers: the price of the book and either a 0 or a 1 (0 if the book is a paperback, 1 if the book is hardback.) Write a program to obtain the following information:

a. Total sales.
b. Total number of books sold.
c. Average price per book.
d. Minimum price of a hardback book.
e. Average price of a paperback book.

28.5. Mrs. Spander is spending her money faster than she earns it. She has now decided to keep track of all her expenses. For every purchase or expense, she enters on her home computer the expense description, the amount, and an expense category code (1 = household, 2 = medical, 3 = recreation, 4 = utilities). Then every Sunday night she runs her budget program to obtain an analysis like the following one:

Budget Analysis

	Household Expenses	Medical Expenses	Recreation Expenses	Utility Bills
Plants	12.33			
Movies			6.50	
Mortgage	389.75			
Dentist		154.25		
Gas				99.66
Vacation			1156.56	
Subtotals	402.08	154.25	1163.06	99.66

Total Expenses $1819.05

All the Best

•••