RECITATION 8

Q1. Write a program with functions.

- In the main function, a table of 10 x 10 is defined.
- The first function (FillMatrix) fills the matrix as shown below.
- The second function (PrintMatrix) prints the content of the matrix.

```
 1   2   3   4   5   6   7   8   9  10
 2   4   6   8  10  12  14  16  18  20
 3   6   9  12  15  18  21  24  27  30
 4   8  12  16  20  24  28  32  36  40
 5  10  15  20  25  30  35  40  45  50
 6  12  18  24  30  36  42  48  54  60
 7  14  21  28  35  42  49  56  63  70
 8  16  24  32  40  48  56  64  72  80
 9  18  27  36  45  54  63  72  81  90
10  20  30  40  50  60  70  80  90 100
```

Q2. Write a program with functions that first reads 2 x 10 integers and stores them on the first 2 rows of a matrix. In the third row, the sum of the corresponding numbers in the first 2 rows is stored. Finally, the full matrix is printed. Use the functions ReadRow, Calculate and PrintMatrix.

```
Enter 2 x 10 integers:
1 2 3 4 5 6 7 8 9 10
1 2 2 1 3 4 1 1 2 1

Table:
   1   2   3   4   5   6   7   8   9  10
   1   2   2   1   3   4   1   1   2   1
   2   4   5   5   8  10   8   9  11  11
```

Q3. Write a program that reads a square matrix, calculates its transpose and prints both the original and the transposed matrix. The matrix cannot be larger than 10 x 10. The wanted dimension is read at the start of the program. Reading the matrix elements, transposing the matrix and printing the matrix is done in 3 different functions.

Q4. Write a program with functions that:

- declares an array with dimensions 10 x 10 in the main,
- uses a function to read the array elements,

- uses a function to find the minimum and maximum element in the array and swaps them (inside the array, using an auxiliary variable is allowed),
- uses a function to print the array after swapping.

Q5. Write a program with a main and 2 extra functions:

- An array of 10 strings is declared in the main function.
- The first function reads the 10 names and stores them in the array.
- The second function prints the names in the array to the screen.

Q6. Write a program with a main and 2 extra functions:

- An array of 10 strings is declared in the main function.
- The first function reads names until the word “end” or the maximum of 10 names is entered and stores them in the array.
- The second function prints the names in the array to the screen.

Q7. Write a program that reads a letter, converts it to Morse code and prints the result to the screen. You can use following array that contains the Morse codes for the letters A, B, C, … consecutively:

*const char *morse[]={".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..",".---", "-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", "...", "-", "..-", "...-", ".--", "-..-", "-.--", "--..", NULL};*

After that, write a program that reads a word and prints it in Morse code to the screen. You can use the function to write the Morse code letter by letter to the screen. For example;

Enter a word: bread

Morse code: -... .-. . .- -..

Q8. Write a program to process test results. To guarantee the product quality, a company takes a sample of N finished parts (0 < N ≤ 20) and submits them to a series of M tests (0 < M ≤ 10). If a part fails one or more tests, data is sent to the computer in the format:

PartNumber TestNumber Result

where 1 ≤ PartNumber ≤ N and 1 ≤ TestNumber ≤ M and Result = 1 for a small error, Result = 3 for a fatal error.

A part is rejected if at least 1 fatal error or at least 3 small errors have occurred. The program first reads the amount of parts tested (N) and the number of tests executed

per part (M). Then, the test results for all failed parts are read until 0 0 0 is entered. The program prints:

- a table with a line for every tested part (so also the ones that did not fail any test) containing information on the test results and a final assessment
- a second table with a line per test containing the number of parts that did not fail, the number of parts that showed a small error and the number of parts that showed a fatal error.

The program uses functions:

- a function ReadNumber that is used twice. Once to read the number of parts tested (N) and once to read the number of tests (M). Make sure only valid numbers are accepted.
- a function to read all test data
- a function to print the results per part
- a function to print the results per test

Hint: Use a matrix of 20 x 10 of which the upper left corner of N rows and M columns is used. Put every entry in the right place in that matrix. The first table to be printed is then just a printout of the used section of the matrix accompanied by some text. The second table can be constructed by counting the occurrences of the numbers 0, 1 and 3 in every column of the matrix.

The screen dialog should look like:

```
How many parts did you test? 5

How many tests did you run? 3

Enter the test results (end with 0 0 0):
4 1 1
5 3 3
3 2 3
2 3 1
3 3 1
4 3 1
4 2 1
0 0 0

Results per part:
part number    tests     assessment
               1  2  3
-------------------------------------
1              0  0  0     accepted
2              0  0  1     accepted
3              0  3  1     rejected
4              1  1  1     rejected
5              0  0  3     rejected

Results per test:
test          failures
        none  small fatal
---------------------------
1       4     1     0
2       3     1     1
3       1     3     1
```