

On the **front of each page, top-right corner**, write your **family name, firstname(s), group number and letter code** – the latter in a circle

P1[40%]

A number of matrices are given in a file named `data.txt`. Matrices are made of strictly positive integers. Write a C program which finds the maximum and the minimum elements in all the matrices supplied in the file and prints its value and the number of the matrix where it occurs last time (first matrix is matrix 1):

Data given as:

- number of matrices (3 in example)
- rows and columns of matrix 1 (3 2 in example)
- elements of matrix 1 (row by row, separated by spaces)
- rows and columns of matrix 2 (2 5 in example)
- elements of matrix 2 (row by row, separated by spaces), and so on

Use as little memory as possible.

Sample input	Output for sample input:
<pre>3 3 2 1 3 17 29 11 5 2 5 9 8 7 6 11 31 67 102 300 1 3 4 8 7 6 11 102 300 1 17 41 43 45 29</pre>	<pre>Max=300 in matrix 3 Min=1 in matrix 3</pre>

P2[60%]

The file named `data.txt` contains only positive integers. You have to print all the integers in that file which have the same sum of the digits as any of the first 60 Fibonacci numbers (The 60-th Fibonacci number is 1,548,008,755,920). Fibonacci numbers are generated using the formula:

$$F_n = F_{n-1} + F_{n-2} \quad \text{where } F_0 = 0, \quad F_1 = 1$$

The data in the file is organized as follows:

- first line: number of values
- next lines: values to check

Sample input
<pre>10 331 6100 123 99999999 87878787 39088169 217839 408320 51449 879</pre>
Output for sample input:
<pre>331 has the sum 7 same as fib[9]=34 6100 has the sum 7 same as fib[9]=34 39088169 has the sum 44 same as fib[38]=39088169 217839 has the sum 30 same as fib[32]=2178309 408320 has the sum 17 same as fib[11]=89 51449 has the sum 23 same as fib[29]=514229 879 has the sum 24 same as fib[16]=987</pre>

On the **front of each page, top-right corner**, write your **family name, firstname(s), group number and letter code** – the latter in a circle

1. (1p) In what way(s) can a **struct** be passed as a parameter to a function in C? Provide short example(s).

2. (2p) Write a recursive C function which takes a one-dimensional array of integers, and the number of its elements, and prints the sum of the odd numbered values. The first value in the array is odd-numbered. E.g., for argument {8,7,6,5,4,3,2,1} printout should be the value of the sum 8+6+4+2, i.e. 20

3. (2p) Trace the program below and show what is printed.

```
#include <stdio.h>
int mystery( int i ) {
    int j = 0 ;
    printf ( "%d\n" , i ) ;
    if ( i > 1 ) j = mystery(i - 1) + 1;
    else j = i;
    printf ( "%d\n" , j ) ;
    return j;
}
int main( ) {
    mystery ( 3 );
    return 0;
}
```

4. (4p) What does the following program print? Find this by determining the contents of variables at each step.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>
int main(void) {
    char *a;
    char *b;
    char c = 'X';
    a = (char *)calloc(8, 1);
    strcpy(a+1, "xyDEFg");
    printf("1: %c\n", a[0]);
    *a = 'A';
    printf("2: %s\n", a);
    a[4] = c;
    b = &c;
    printf("3: %c\n", *b);
    b = &(a[3]);
    a[1] = toupper(a[1]);
    a[2] = toupper(a[2]);
    printf("4: %s\n", a);
    return 0;
}
```
