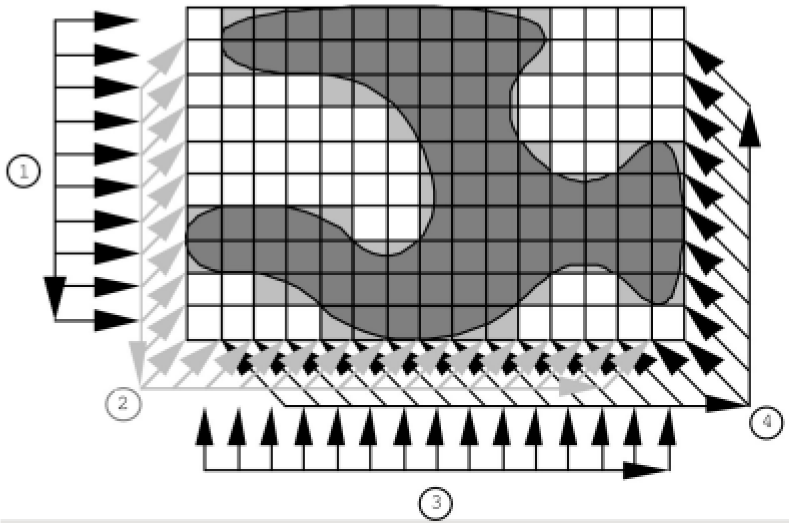
2nd Set of Lab Exercises

Part 1



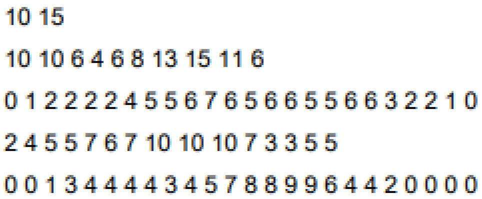
A body scanner operates with the help of sensors that detect the mass density along linear axes. The values recorded by the sensors are then processed to produce an image representing the scanned body. The object of this exercise is to build a program that will perform the above processing. The program will accept as input a file with the values recorded by the sensors and will produce as output another file that corresponds to the image resulting from the processing. Assume that the scanner consists of four rows of sensors placed around a grid of dimensions N X M, as shown in the figure alongside, where N and M have values of 10 and 15 respectively. The row of sensors (1) consists of 10 sensors that “see” horizontally from left to right. Row (2) has 24 sensors that “see” diagonally from bottom left to top right. Row (3) has 15 sensors that “see” vertically from bottom to top. Finally, row (4) has 24 sensors that “see” diagonally from bottom right to top left. During a scan, each sensor records the density of the object along the corresponding line on the grid. Specifically, each sensor scans a “ray” and essentially measures in how many cells of the grid the density is high (value 1), and in how many it is low (value 0), that is, it sums up the aces along the scan line of the black and white image. Therefore, the values that arise after a scan are as many as the scan “rays”. The reading of the values given by the sensors is done in a counterclockwise direction. A full scan consists of 73 values.

Write a program in C language that will read a file containing the values recorded by the sensors, produce the image corresponding to the input values, and print it to a file.

NOTE: In the general form of the problem, there may be cases of ambiguity where the values of the sensors are not sufficient to reconstruct the image. Assume that your program does not need to deal with these cases.

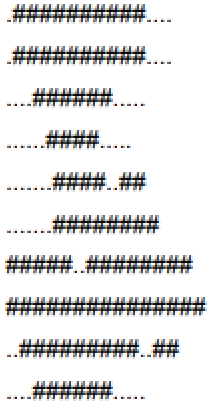
Input file

The input file starts with two numbers corresponding to the values of N and M. It then contains the values recorded by the sensors in row (1), then row (2), row (3), and row (4) at the end. For the example of the image in the previous figure, the input file is as follows:



```
10 15
10 10 6 4 6 8 13 15 11 6
0 1 2 2 2 2 4 5 5 6 7 6 5 6 6 5 5 6 6 3 2 2 1 0
2 4 5 5 7 6 7 10 10 10 7 3 3 5 5
0 0 1 3 4 4 4 4 3 4 5 7 8 8 9 9 6 4 4 2 0 0 0 0
```

Output file
The output file will correspond to a black and white image. The pixels representing body areas (dark in the example in the figure) are displayed with the character '#'. The remaining pixels (white in the
example in the figure) are displayed with the character '.'
For the example image the output file will have the form:

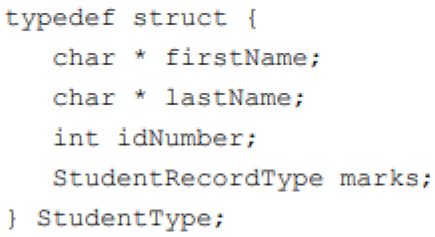


```
.##########....
.##########....
....######.....
......####.....
.......####..##
.......########
#####..########
###############
..#########..##
....######.....
```

ATTENTION: Your program should perform all necessary checks to ensure its proper operation.

Part 2

Design and implement a program in the C programming language that will manage a small database with the students of a class and their performance in the subjects. Each student will be modeled with a structure (struct) in the following format:



```
typedef struct {
   char * firstName;
   char * lastName;
   int idNumber;
   StudentRecordType marks;
} StudentType;
```

Each student's "folder" with the grades of the first and second exercises, progress, and final exam is modeled in a structure as below:

```
typedef struct {
   float firstAssignment;  /* Ο βαθμός της πρώτης άσκησης */
   float secondAssignment; /* Ο βαθμός της δεύτερης άσκησης */
   float midterm;          /* Ο βαθμός της προόδου */
   float final;            /* Ο βαθμός του τελικού διαγωνίσματος */
   float finalMark;        /* Ο γενικός βαθμός για το μάθημα  */
} StudentRecordType;
```

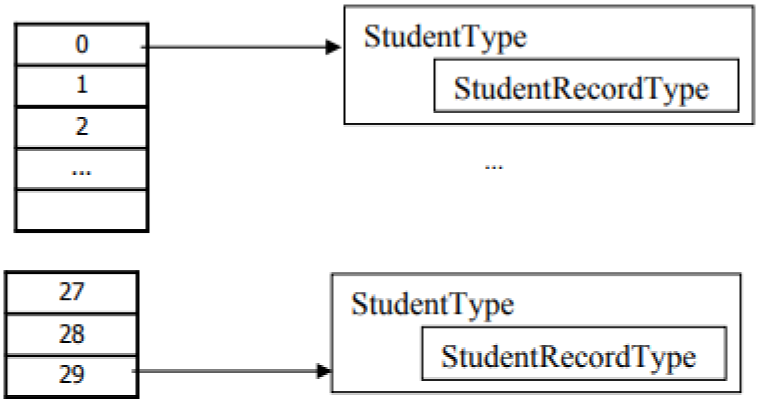
Assume that for this hypothetical class, the first assignment is worth 10% of the grade, the second assignment is worth 15% of the grade, the progress is worth 25% of the grade, and the final exam is worth 50% of the grade. All grades are non-negative with a maximum of 100. Assume that the class has a maximum of 30 students. Your program will read a file named project2-askhsh2.dat of the form:

```
<πλήθος μαθητών>
<όνομα> <επώνυμο>
<βαθμός πρώτης άσκησης>
<βαθμός δεύτερης άσκησης>
<βαθμός προόδου>
<βαθμός τελικού διαγωνίσματος>
<όνομα> <επώνυμο>
<βαθμός πρώτης άσκησης>
<βαθμός δεύτερης άσκησης>
<βαθμός προόδου>
<βαθμός τελικού διαγωνίσματος>
...............
END
```

π.χ.

```
10
Yiannis Yiannopoulos
90
78
67
80
Kostas Kontogiannis
50
60
40
90
....
END
```

Graphically, the class structure is as shown in the figure below:



Implement a program that:
1. Reads the data
2. Calculates the final grade of each student
3. Calculates the class average

4. Finds and prints a student's file based on their last name
5. Prints the results alphabetically in the following format:

```
RESULTS
-----------------------
DETAILED
<name> <surname> <final grade>
<name> <surname> >final grade>
.....
CLASS AVERAGE
<average>
```

Each of the above functions 1-5 should be implemented as a separate function, The base should be implemented as a vector of 30 indicators in StudentType type data structures. The required number of StudentType type data structures will be created after reading the number of students from the file project2-askhsh2.dat.