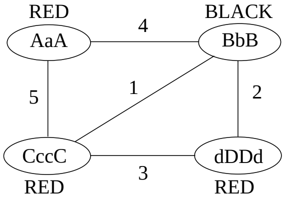
Algorithms and Programming
22 February 2016
Part II: Program (18 point version)

No books or notes are allowed. Examination time: 100 minutes.
The final program is due: before Tuesday the 23rd, 12.00 p.m. for all students who wants to graduate in the March session, and before Thursday the 25th, 12.00 p.m. for all other students.
Send the program by e-mail to `stefano.quer@polito.it`.

A colored weighted undirected graph G is stored on a file, with the format:
`idV1 colV1 val idV2 colV2`
where `idV1` is the identifier of the first vertex of color `colV1`, `idV2` is the identifier of the second vertex, of color `colV2`, and adjacent to the first one, and `val` is the weight of the edge connecting `idV1` to `idV2`. The number and the order of the vertices within the file is unknown. Each vertex identifier `idV` is a string of at most 20 characters. The color of a vertex can either be equal to `RED` or `BLACK`.
The following is a correct example of such a file, with the corresponding graph representation.

```
AaA RED 4 BbB BLACK
BbB BLACK 1 CccC RED
AaA RED 5 CccC RED
dDDd RED 3 CccC RED
BbB BLACK 2 dDDd RED
```



Write a C program able to:

- Receive the file name on the command line.
- Store the graph in a proper data structure.
- Solve the following two distinct problems:
  - Find a simple path of maximum weight such that each BLACK vertex is followed by a RED or BLACK vertex, whereas each RED vertex is strictly followed by a BLACK vertex. Once found such a path, print it out on standard output.

    In the previous example graph, such a path may be (`AaA`, `BbB`, `dDDd`), with weight equal to $4 + 2 = 6$.
  - Find the connected sub-graph whose sum of the edges' weight is maximum and the number of RED or BLACK vertices differ at most of two. Once found such a graph, print the list of vertices out on standard output.

    In the previous example graph, such a graph is the graph itself, i.s., (`CccC`, `AaA`, `BbB`, `dDDd`), with total weight equal to $5 + 4 + 1 + 2 + 3 = 15$.

Algorithms and Programming

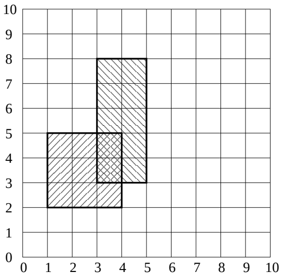
22 February 2016

Part II: Program (12 point version)

No books or notes are allowed. Examination time: 100 minutes.
The final program is due: before Tuesday the 23rd, 12.00 p.m. for all students who wants to graduate in the March session, and before Thursday the 25th, 12.00 p.m. for all other students.
Send the program by e-mail to `stefano.quer@polito.it`.

1 (2.0 points)
A file stores an undefined number of rectangles, with their sides parallel to the Cartesian axes, and placed within the area in the Cartesian plane defined by the points (0, 0) and (10, 10). Each rectangle is defined by specifying its bottom-left and top-right vertices. For example, the file reported on the left-hand side of the following picture, defines two rectangles represented on the right-hand side:

```
1 2 4 5
3 3 5 8
```



Write function:
`int area (FILE *fp);`
receiving an open file, with the previous format, and returning the total area covered by the rectangles taking into consideration overlapping areas. For example, for the previous picture the function should return an area value equal to 17.

2 (4.0 points)
The `list_t` type points to a list of integer values. Write function:
`void split (list_t **p, int threshold, list_t **p1, list_t **p2);`
able to split such a list into two similar lists based on the `threshold` value. The function receives (by reference) the pointers to the head of the input list `p`, and the resulting lists `p1` and `p2`. It extracts all elements from `p`, and in `p1` all elements smaller than the threshold. and in `p2` all elements larger or equal to the threshold.
The order of the elements within the original list `p` has to be maintained into `p1` and `p2`. At the end of the function `p` must be empty. Notice that it is requested to write the code for the entire `split` function, without using library functions to manipulate lists.
For example if `p` stores elements (`7, 8, 25, 2, 9, -5, 10, 37`), and the `threshold` is 18, then `p1` should contains (`7, 8, 2, 9, -5, 10`), and `p2` (`25, 37`).

3 (6.0 points)
A set of bulbs are turned-on and off by a set of switches. Initially, all bulbs are turned off. Each switch is connected to some set of bulbs; pressing the switch will turn each of them on if they are off, and will turn each of them off if they are on. An electrician would like to understand which is the smaller set of switches (if it exists) that have to be pressed to turn-on all the bulbs.
A matrix stores the "electrical connections", i.e., the set of bulbs that can be turned on and off by each switch. For example, for 4 switches (rows) and 5 bulbs (columns), the matrix:

	0	1	2	3	4
0	1	1	0	0	1
1	1	0	1	0	0
2	0	1	1	1	0
3	1	0	0	1	0

indicates that the first switch (switch number 0) turns on and off bulbs 0, 1, and 4. In this case the minimum number of switches able to turn on all the bulbs is 3 with switches (0, 1, 3).
Write a function able to receive the matrix and all other data deemed necessary as parameters, and to print out the solution to the puzzle (i.e., the switches to press), if one solution exists, on standard output.