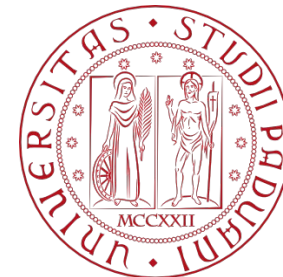


COMPUTER ENGINEERING LABORATORY

Luigi Rizzo

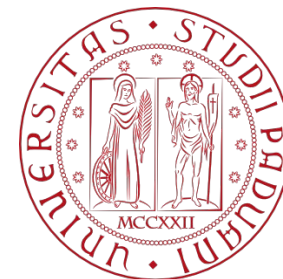
luigi.rizzo@unipd.it

October 2024-January 2025



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Exercises: test simulation



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Exercise 1



A file, (probably?) present on disk, contains a sequence of words, each with a maximum length of 32. The words can contain lowercase and uppercase alphabetic characters, punctuation, apostrophes, brackets and are separated from each other by spaces and/or newline.

Write a C program that takes the name of the file as an argument and returns (via data chosen by the student) the number of occurrences of the lowercase characters making up only words at least 8 characters long.

Exercise 1



For example, considering the following file:

this file
contains at least
three significant words
that are sufficiently long

the program shall return to the calling program and in the way chosen by the student the following information

Exercise 1



There are 2 occurrences of the character a
There are 3 occurrences of the character c
There are 1 occurrences of the character e
There are 3 occurrences of the character f
There are 1 occurrences of the character g
There are 6 occurrences of the character i
There are 1 occurrences of the character l
There are 5 occurrences of the character n
There are 1 occurrences of the character o
There are 3 occurrences of the character s
There are 3 occurrences of the character t
There are 1 occurrences of the character u
There are 1 occurrences of the character y

<a:2> <c:3> <e:1> <f:3>
<g:1> <i:6> <l:1> <n:5>
<o:1> <s:3> <t:3> <u:1>
<y:1>

Exercise 2

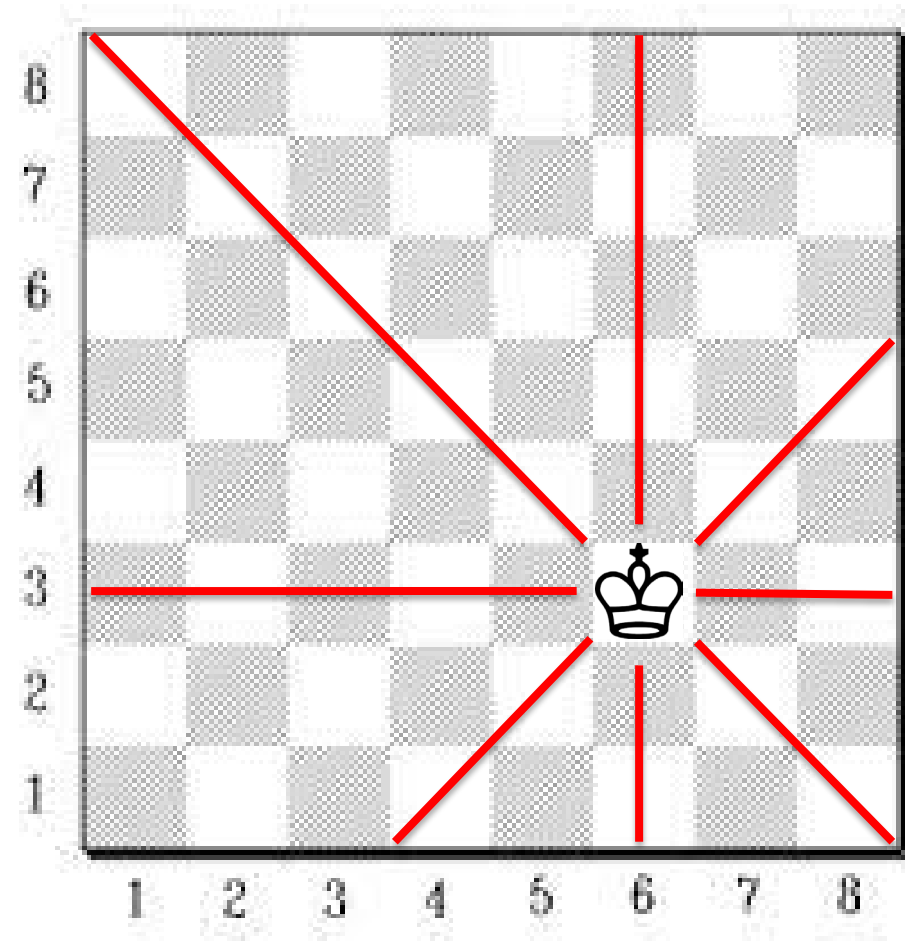


- Two pieces are placed on an 8x8 chessboard: the white King and the black Queen.
- Write a program in C language which, having acquired the positions of the King and the Queen, determines whether the Queen is in such a position as to be able to eat the King. The positions of the two pieces are identified by the row and column on which they are located , expressed as integers between 1 and 8.

Some suggestions:

- Define all the situations when the Queen eats the King (same row, same column, ascending diagonals, descending diagonals)
- Solve one subproblem at a time and check the solution before proceeding with the next subproblem.

Exercise 2



Exercise 3



Consider the following declarations of types and variables, which define the data structures to represent meteorological surveys carried out for 100 municipalities throughout the year.

Write a code fragment, defining any additional variables, which finds all the cities that in the month of December recorded at least one survey in which the level of rainfall was higher than that stored in the threshold variable **thresholdF**. For each of these cities, display on a single line the name and the average amount of rainfall calculated over the entire year.

Exercise 3



```
typedef struct {  
    unsigned int day; /* between 1 and 31 */  
    unsigned int month; /* between 1 and 12 */  
    double levelF; /* rain fall */} measure;
```

```
typedef struct {  
    char municipality[32]; /*name of municipality*/  
    measure surveys[366]; /* max 366 findings, it is not certain that surveys take place on a daily  
basis */  
    int nSurveys; /* actual number of data recorded in surveys */} measuresMunicipality; /* data  
monitored during the year for a single municipality */
```

Exercise 3



```
typedef measuresMunicipality surveysGlobal[100]; /* data monitored during the year for all  
100 municipalities */  
surveysGlobal dg;  
float thresholdF; /* Assume that the variables thresholdF and dg have been initialized via  
instructions not shown here */
```

Exercise 3



```
typedef struct {  
    unsigned int day;  
    unsigned int month;  
    double levelF;  
} measure;
```

```
typedef struct {  
    municipality char[20];  
    measure surveys[366];  
    int nSurveys;  
} measuresMunicipality;
```

```
typedef measuresMunicipality surveysGlobal[100];  
surveysGlobal dg;  
double thresholdF;
```

Exercise 3



Consider the file `Daily_Rainfall_2020_Multiple_Cities.csv` available in moodle.

Having the following format

`Date;Los Angeles (mm);Chicago (mm);Houston (mm);Phoenix (mm);New York (mm)`

`01/01/2020;1,3361387227156252;3,1123791335717077;4,874491512053722;0;3,9934283060224653`

as input file for your program.

Sum of Digits

Write a recursive function to calculate the sum of digits of a given integer.

Hint: For an integer n , the sum of digits is $n \% 10 + \text{sumOfDigits}(n / 10)$.

The base case is when n becomes 0.

Reverse a String

Write a recursive function to reverse a string.

Hint: Use two pointers, one at the beginning and one at the end of the string and swap the characters they point to. Then move the pointers inward and recursively reverse the remaining substring.

Check if a String is a Palindrome

Write a recursive function to check if a given string is a palindrome.

Hint: Compare the first and last characters of the string. If they match, recursively check the substring that excludes the first and last characters.

GCD of Two Numbers

Write a recursive function to find the Greatest Common Divisor (GCD) of two numbers.

Hint: Use the Euclidean algorithm: $\text{gcd}(a, b) = \text{gcd}(b, a \% b)$ with the base case $\text{gcd}(a, 0) = a$.