

**Project 1: Decimal - to – Binary – to – Hexadecimal Converter**

Write a program that performs conversions between natural numbers (in decimal form) its binary form and hexadecimal form. It should have GUI (Graphical User Interface).

**Project 2: Sorting** (the algorithm will be provided)

Write a program that will sort all the integer numbers stored in a file in *decreasing order*, using the algorithm given below. Please, note that you cannot use Python's libraries or built-in functions to do the sorting. You should use the algorithm given to you. The numbers will be read from a file, and the sorted numbers should be output into another file. The numbers in the file can be separated by a space or a new line characters (you will have to take care of both cases).

Here is an example of the content of the input file:

```
12 76 1 0 45
23 5 9
1 2
```

**Project 3: Counting numbers program**

Write a program that reads-in a list of integers from a file (the file name is given by the user) and stores them into a list, say A. The output is to be a two-column list. The first column is a list of distinct list elements; the second column is the count of the number of occurrences of each element. The two-column list should be sorted on entries in the first column, largest to smallest. The result should be displayed in Python shell and stored in an output file. The user should be notified about the name of the output file.

example:

Input data (taken from a file): 2 6 3 8 1 3 6

Output (to Python shell and to a file):

integer values	occurences
8	1
6	2
3	3
2	1
1	1

**Project 4: Counting letters program**

Write a program that finds the number of occurrences of letters from the alphabet (ignoring lower case – upper case) in a file (the file name is given by the user), then outputs the two-column list with result. The first column is a list of letters of the alphabet; the second column is the count of the number of occurrences of each letter from the alphabet. The two-column list should be sorted on entries in the first column, i.e. it starts from letter 'a' and ends with the letter 'z'.

example:

Input data (taken from a file): Hello, my name is Natalia Novak.

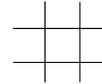
Output:

letters of the alphabet	occurences
a	5
...	# skipping few letters
h	1
i	2
j	0
k	1
l	3

**Project 5: Tic-Tac-Toe game**

Write a program that will allow two users to play tic-tac-toe using graphics. Player vs computer. The program should ask the player whether he wants to play Xs or Os. X goes first. The object of Tic Tac Toe is to get three in a row.

At the beginning the program displays the empty tic-tac-toe board:



Then each player to make a choice (X or O), computer will play the other. The program should notify the user when is his/her turn. After each click, the program displays the changed board. The program should not allow the user to click in occupied slots.

**Project 6 (group project - two people):**

*Statistics program - Regression line or "Line of Best Fit"* (using graphics)

Write a program that reads a data set of two-dimensional points and calculates the "line of best fit" for that point set. Plot the data and resulting line graphically.

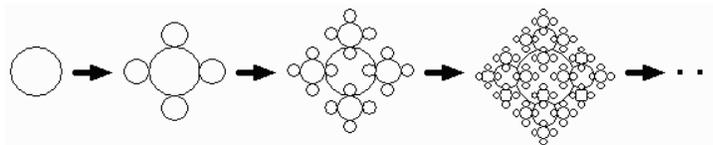
1<sup>st</sup> participant: deals with the plotting the data,

2<sup>nd</sup> participant: calculates the line of best fit (regression line) and plots the resulting line of best fit

**Project 7: Fractals drawings (using recursion)**

Fractals can be understood as the result of iterating geometric constructions. For example, if we use a geometric construction:

1. Start with a circle,
2. Draw circles of half the previous diameter at 12 o'clock, 3 o'clock, 6 o'clock and 9 o'clock on the circle that you have just finished,
3. Repeat step 2,

**Project 8 (group project): Matrix arithmetic (using file input and graphics)**

Design and implement program that will allow the user to play with matrix operations (addition, multiplication and transpose of a matrix).

Input: two matrices (from files)

The rest: in the graphics window (file name should also be collected in the graphics window)

1<sup>st</sup> participant: writes functions of acquiring of matrices from the file, addition of two matrices, multiplication of two matrices and of transpose of a matrix.

2<sup>nd</sup> participant: writes functions of screen work (displays matrices, operations control (addition, multiplication and transpose), and displays the result)

**Project 9: Image processing** (using graphics)

Write a program that converts a color image to gray-scale and to its color negative. At the click of the mouse, the program performs conversion, displays the images and then prompts the user for two file names to store the gray-scale and color negative images.

**Project 10: Bar chart of student's exam scores**

Write a program to plot a horizontal bar chart of student exam scores.

The exam scores are stored in a file (separated by either space or next line).

The program should work 100% in graphics window, meaning that the file name should also be requested in the graphics window.

Draw the scale. Mark the bars with the score.

**Project 11: Bouncing smiley face** (using graphics)

Write a program to animate a smiley face bouncing around a window.

Every time the program is run, the smiley face should start at a random place inside the window.

Exit button should be present to terminate the program.

**Project 12: Bit Strings** (Discrete Math)

Write a program to find the bitwise *AND*, bitwise *OR* and bitwise *XOR* of two bitstrings of length  $n$ .

The program should be working fully in graphics window.

Comments:

- bitstrings will be provided by the user (from a file or from a keyboard, in the graphics window - your choice),
- the results (along with the provided bitstrings) should be displayed in a graphics window.

**Project 13: Set operations** (Discrete Math)

Write a program that given two sets **A** and **B** (from the same universal set **U**), finds the union **A∪B**, their intersection **A∩B**, their difference **A-B**, and their Cartesian product **A×B**.

Do not use the built-in set type.

Use graphics window to get the sets A and B and to display the results. You may have five separate buttons (for union, intersection, difference, Cartesian product, and Exit button).

**Project 14:** *Find the change to  $n$  cents using quarters, dimes, nickels and pennies (Greedy Algorithm)*  
Write a program that given an integer  $n$ , uses greedy algorithm to find the change for  $n$  cents using quarters, dimes, nickels and pennies. Everything should be done in graphics window.

**Project 15:** *Rock-Paper-Scissors Game*

Write a program that lets the user play the game of Rock, Paper, Scissors against the computer, using graphics.

**Project 16:** *Driver's Licence Exam*

The local driver's license office has asked you to create an application that grades the written portion of the driver's license exam. The exam has 20 multiple-choice questions. Here is an example of the ten correct answers (in a file):

1. B   6. A   11. B   16. C  
2. D   7. B   12. C   17. C  
3. A   8. A   13. D   18. D  
4. A   9. C   14. A   19. D  
5. C   10.D   15. D   20. A

Comments:

- Your program should work in graphics window
- Two input files: file with correct answers, and a file with student's answers
- The program should display a message indicating whether the student passed or failed the exam. (A student must correctly answer 15 of the 20 questions to pass the exam.) ,
- It should also display the number of correctly answered questions, the percentage of the correctly answered questions of the total number of questions, and a list showing the question numbers of the incorrectly answered questions.