Study year: 2021/2022

Group 1

Midterm 2

Computer Practicum 1

Instructions:

- The midterm exam contains three assignments that are worth a total of 100 points.
- You have 90 minutes to solve the tasks.
- The use of literature, the Internet and other media are NOT permitted.
- The use of notes on one A4 sheet (written by hand) is allowed.
- Prepare the solution of each task in a (separate) file.
- Solutions (3 C scripts) will be submitted via the e-classroom after the writing time according to the instructions you will receive from the TA.

Task 1 - Easy (25 points)

World War Z ★

Produce the shape below given the following value of positive integer n ≥ 3.

$n = 3$	$n = 5$	$n = 6$

`ooo`
` o&nbsp;`
`ooo`
	`ooooo`
	`   o&nbsp;`
	`  o&nbsp;&nbsp;`
	` o&nbsp;&nbsp;&nbsp;`
	`ooooo`
		`oooooo`
		`    o&nbsp;`
		`   o&nbsp;&nbsp;`
		`  o&nbsp;&nbsp;&nbsp;`
		` o&nbsp;&nbsp;&nbsp;&nbsp;`
		`oooooo`

Required components to implement:

- loops and conditional constructs
- proper variable names and comments
- input validation (accept correct inputs and reject wrong inputs)

Task 1 - Easy (25 points)

Tournament Matches ★

Write a program that asks the user how many teams are participating in the tournament. Let us assume that the teams are numbered starting from 1, 2, 3, and so on. In the tournament, each team has to face every other team **exactly once**. The program should display all the matches to be played. Assume that there are at least 2 teams Refer to the example below:

$number\ of\ teams = 2$	$number\ of\ teams = 3$	$number\ of\ teams = 4$

`Team 1 vs. Team 2`	`Team 1 vs. Team 2`
	`Team 1 vs. Team 3`
	`Team 2 vs. Team 3`
		`Team 1 vs. Team 2`
		`Team 1 vs. Team 3`
		`Team 1 vs. Team 4`
		`Team 2 vs. Team 3`
		`Team 2 vs. Team 4`
		`Team 3 vs. Team 4`

Required components to implement:

- loops and conditional constructs
- proper variable names and comments
- assume the number of teams is even. however you get higher points if you can handle an odd number of teams.
- input validation (accept correct inputs and reject wrong inputs)

Task 1 - Easy (25 points)

Alternating ★

Given an integer n, we can print the numbers from 1 to n. We can also print the numbers from n to 1. But for this problem, you have to do both of those things alternately, **at the same time**! For example, if the input for n is 10, the output should be in this order:
$1,\ 10,\ 2,\ 9,\ 3,\ 8,\ 4,\ 7,\ 5,\ 6,\ 6,\ 5,\ 7,\ 4,\ 8,\ 3,\ 9,\ 2,\ 10,\ 1$

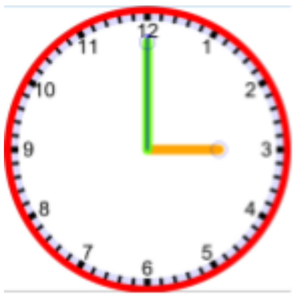
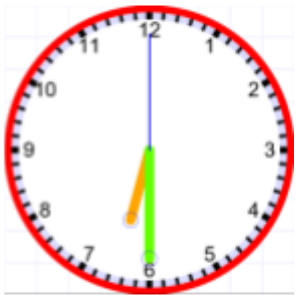
Required components to implement:

- loops and conditional constructs
- proper variable names and comments
- assume n is even.
- input validation (accept correct inputs and reject wrong inputs)

Task 2 - Intermediate (35 points)

Analog Clock Blues ★ ★

Write a program that asks for the number of hours and the number of minutes. Output both angles (in degrees) between the hands of an analog clock if this time were displayed. For example, if the number of hours is 12 and the number of minutes is 0 (3:00), the output should be 90 degrees and 270 degrees. If the number of hours is 6, and the minutes is 30 (6:30), the output should be 15 degrees and 345 degrees. Refer to the illustrations.

```
Input number of hours: 3
Input number of minutes: 0
The angle between the hands are 90 and 270 degrees.
```

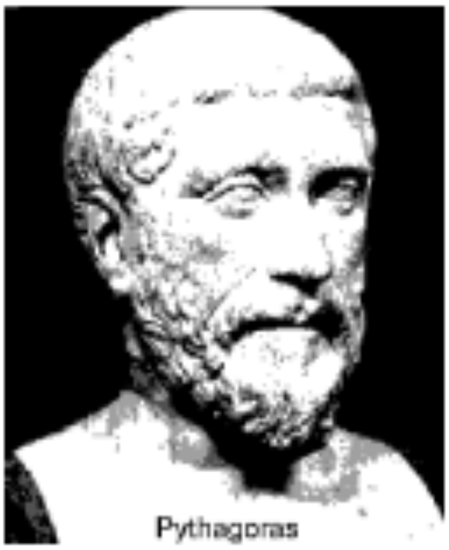
Required components to implement:

- user-defined functions
- parameter passing by value
- proper variable names and comments
- input validation (accept correct inputs and reject wrong inputs)

Task 2 - Intermediate (35 points)

Perfect Numbers ★★

The discovery of perfect numbers was lost in prehistory, but it was known that the Pythagoreans (founded 525 B.C.) studied them for their mystical properties. The mystical tradition was continued by the Neo-Pythagorean philosopher Nicomachus of Gerasa. He gave moral qualities to his definitions, and such ideas found credence among early Christian theologians. Often the 28-day cycle of the Moon around the Earth was given as an example of a "Heavenly," hence perfect, event that naturally was a perfect number.



A perfect number is a positive integer which is equal to the sum of its proper divisors. A proper divisor means all of its divisors except itself. The first perfect number is 6, which has the proper divisors 1, 2, and 3. The next perfect number is 28, with proper divisors 1, 2, 4, 7, and 14.

While most people do not believe in the mystical properties of these numbers today anymore, perfect numbers are still studied in mathematics because they are cool. Write a program that accepts a positive integer. The program should then output if it is a perfect number or not

Required components to implement:

- user-defined functions
- parameter passing by value
- proper variable names and comments
- input validation (accept correct inputs and reject wrong inputs)

Task 2 - Intermediate (35 points)

Eidolon Summoning ★★

Eidolons are mythical creatures that visit the land of the living. Each Eidolon has the ability to grant a wish on certain aspects or realms of human lives. Humans are eager to predict the year a specific Eidolon will appear to prepare for its coming. Your task is to write a program that teaches about the different years of visitation from various Eidolons. You must carefully look at the following chart to determine your program behavior:

Name of Eidolon	**Years of Visitation**	**Blessing**
Phoenix	Every 6 years	Health
Bahamut	Every 5th decade	Wisdom
Leviathan	Every 3rd visit of Phoenix for the century	Peace

We will assume that Phoenix visits in years 6, 12, 18, and so on. Bahamut visits in years 50, 100, 150, and so on. Leviathan visits in year 18, because it is the third visit of the Phoenix for that century (1-100). Its next visit will be in year 114.

Your program should accept the current year (assume that the user inputs a year from 1 to 9999). **For every** Eidolon that will visit in that year, your program should output a message in the following format:

`<Eidolon name> has come forth! - Wish for <blessing>!`

For example, if the year is 1050, the output of your program should be:

```
Phoenix has come forth! - Wish for health!
Bahamut has come forth! - Wish for wisdom!
```

Required components to implement:

- user-defined functions
- parameter passing by value
- proper variable names and comments
- input validation (accept correct years and reject wrong inputs)

Task 2 - Intermediate (35 points)

Countdown ★ ★

The deadline for the homework is fast approaching! Write a program that accepts the current time (expressed in hours and minutes) and the time of the deadline (expressed in hours and minutes). Both times are expressed as military time (1PM is denoted as 13). The program should display exactly how many minutes are left before the deadline. You may assume that the current time and the deadline time always falls within the same day.

```
Current hour: 13
Current minute: 30
Deadline hour: 14
Deadline minute: 45
You have 75 minutes left!
```

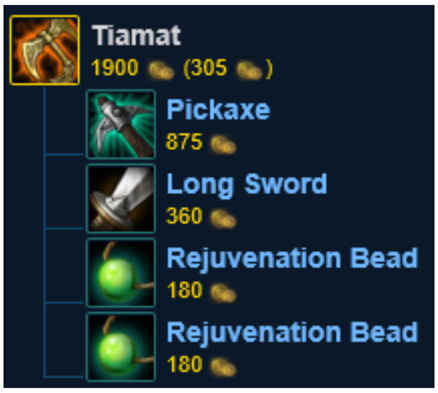
Required components to implement:

- user-defined functions
- parameter passing by value
- proper variable names and comments
- input validation (accept correct inputs and reject wrong inputs)

Task 3 - Advanced (40 points)

Item Crafting ★★★

In a video game called *League of Legends*, one of the items that can be acquired is called the Tiamat. To be able to craft this item, you need to have **one (1) pickaxe**, **one (1) long sword**, and **two (2) rejuvenation beads**.



Write a program that will ask the user how many pickaxes, long swords, and rejuvenation beads he has in his inventory. The program should output the following:

(1) The maximum possible number of Tiamat items that he can craft, and

(2) The remaining number of items of each type after crafting the Tiamat items.

For this problem, assume that the user's inventory has **unlimited capacity** (even if you can only carry 6 items at once in the actual game). Refer to the following sample run:

```
Number of pickaxes: 4
Number of long swords: 5
Number of rejuvenation beads: 18

Maximum number of Tiamat items you can craft: 4
Remaining pickaxes: 0
Remaining long swords: 1
Remaining rejuvenation beads: 10
```

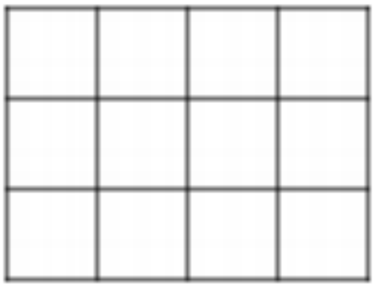
Required components to implement:

- user-defined functions
- parameter passing by value (higher score if we use parameter passing by reference)
- structures/records/array
- proper variable names and comments
- input validation (accept correct inputs and reject wrong inputs)

Task 3 - Advanced (40 points)

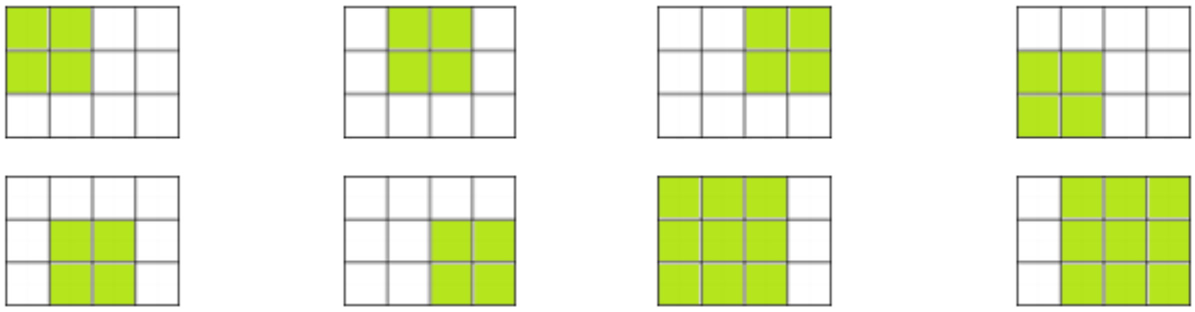
Bathroom Tiles ★★★

Many bathrooms are covered with grids of square tiles. The figure below shows a grid of tiles (3 rows and 4 columns).



Your task is to write a program that will output the total number of squares in the grid of tiles. For a grid of tiles as shown above, the output should be 20.

Yes, you've read that right. The answer is 20, and not 12. As you can see, in addition to the 12 individual squares, there are 8 other squares that span more than one tile, as shown below:



Your program should ask for the dimensions of the grid (number of rows and number of columns). Your program should then output the total number of squares that can be found in the resulting grid.

To help you, here are a few more examples. A 2x2 grid has a total of 5 squares (4 individual squares and one large square). A grid has 5 squares (there are no larger squares aside from the individual ones). A grid has a total of 70 squares.

Required components to implement:

- user-defined functions
- parameter passing by value (higher score if we use parameter passing by reference)
- structures/records/arrays
- proper variable names and comments
- input validation (accept correct inputs and reject wrong inputs)