On-line «Master in Computer Vision» educational program.
Entrance examination test (each problem is evaluated by 10 points)**.**
Variant 3.

1. Find all pairs *(a, b)* of non zero integer numbers *a* and *b* from the interval (-15, 15) , such that, the following system of equations has no solution

$$\begin{cases} ax_1 + bx_2 = 2 \\ bx_1 + 3x_2 = -1 \end{cases}$$

Solution. Under conditions on *a* and *b* (*a* and *b* are non zero) the system of equations is equivalent to the system

$$\begin{cases} 3ax_1 + 3bx_2 = 6 \\ b^2x_1 + 3bx_2 = -b \end{cases}$$

It implies

$$(3a - b^2)x_1 = 6 + b.$$

If $(3a - b^2) \neq 0$ then the system has a unique solution

$$x_1 = \frac{6+b}{3a-b^2}, x_2 = -\frac{1}{3}(1 + bx_1) = -\frac{1}{3}\frac{(3a+6b)}{(3a-b^2)}$$

If $(3a - b^2) = 0$ and $6 + b \neq 0$, then the system has no solution. Corresponding pairs of non zero integers *(a, b)* from the interval (-15, 15) are (3, -3), (3, 3), (12, 6).
If $(3a - b^2) = 0$ and $6 + b = 0$, then $b = -6, a = 12$. The system takes the form

$$\begin{cases} 12x_1 - 6x_2 = 2 \\ -6x_1 + 3x_2 = -1 \end{cases}$$

There is only one equation

$$12x_1 - 6x_2 = 2$$

Therefore, the system of equations has infinitely many solutions (x_1, x_2)
Answer. The system of equations has no solution for the following pairs *(a, b)* of non zero integer numbers *a* and *b* from the interval (-15, 15):

(3, -3), (3, 3), (12, 6).

2. Let *x*,*y*,*z* be boolean variables (take only values 0 and 1). Function *f(x,y,z)* is defined by the following operations in the boolean algebra:

$$f(x, y, z) = (x \land y) \lor (y \land \neg z)$$

where

$\land$ - is the conjunction

$\lor$ - is the disjunction

$\neg$ - is the negation

Define a new function

$$g(x, y, z) = f(\neg y \land z, x, x \lor y)$$

- Find the boolean values of *g(x,y,z)* for all possible boolean values of *x*,*y*,*z* (truth table)
- Find an explicit formula for the function *g(x,y,z)*
- Simplify the formula for *g(x,y,z)* using the boolean algebra laws (each step of simplification is evaluated)

Solution. Direct calculations give the truth table for the function *g(x,y,z)*

x	*y*	*z*	*g(x,y,z)*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Formula for the function *g(x,y,z)*

$$g(x, y, z) = f(\neg y \land z, x, x \lor y) = ((\neg y \land z) \land x) \lor (x \land \neg(x \lor y))$$

Simplification

$$g(x, y, z) = \big((\neg y \land z) \land x\big) \lor \big(x \land \neg(x \lor y)\big) = \big((\neg y \land z) \land x\big) \lor (x \land \neg x \land \neg y) =$$

$$= \big((\neg y \land z) \land x\big) \lor (0 \land \neg y) = x \land \neg y \land z$$

3. Probability distributions of two independent discrete random variables X and Y are given by the tables

X	-1	0
Prob	1/3	2/3

Y	2	3
Prob	1/4	3/4

- Find probability distributions of the random variables X^2 and Y^2
- Calculate the expected value (mathematical expectation) of the random variable $Z=X^2Y^2$
- Calculate the variance of Z.

Solution.

X^2	1	0
Prob	1/3	2/3

Y^2	4	9
Prob	1/4	3/4

$X^2\, Y^2$	0	4	9
Prob	8/12	1/12	3/12

Expectation of Z: $\mathrm{E}(\mathrm{Z}) = \mathrm{E}(\mathrm{X}^2\mathrm{Y}^2) = 4 \cdot \frac{1}{12} + 9 \cdot \frac{3}{12} = \frac{31}{12}$

Variance of Z; $\mathrm{V}(\mathrm{Z}) = \mathrm{E}(Z^2) - \mathrm{E}^2(\mathrm{Z}) = \frac{259}{12} - \frac{961}{144} = \frac{2147}{144}$

$$\mathrm{E}(\mathrm{Z}^2) = \mathrm{E}(\mathrm{X}^4\mathrm{Y}^4) = 16 \cdot \frac{1}{12} + 81 \cdot \frac{3}{12} = \frac{259}{12}$$

$X^4\, Y^4$	0	16	81
Prob	8/12	1/12	3/12

In the following problem it is necessary to suggest the most efficient algorithms. Full points are given for the most efficient algorithm having the lowest computational complexity. The lower the efficiency of the suggested solution, the lower are the points.

4. Write a pseudo-code (or code on any programming language) of an algorithm, which find two elements in an array of real numbers (positive and negative), such that the difference of these elements is maximal. Discuss the computational complexity of your algorithm.

Solution:
Bad:
get_elements_with_max_diff(*array*):
max, *elem*1, *elem*2 = -inf, 0, 0
for *i* = 0, …, |*array*| - 1
for *j* = *0*, …, |*array*| - 1
if *array*[*i*] - *array*[*j*] > *max*
max, *elem*1, *elem*2 = *array*[*i*] - *array*[*j*], *array*[*i*], *array*[*j*]
return *elem*1, *elem*2
The computational complexity of this solution is $O(n^2)$, because there are two nested loops with $O(n)$ iterations.

Normal:
get_elements_with_max_diff(*array*):
sort(*array*)
n = |*array*|
return *array*[*n-1*], *array*[*0*]
The computational complexity of this solution is $O(n\log n)$ because of sort() procedure.

Good:
get_elements_with_max_diff(*array*):
max, *min* = *array*[*0*], *array*[*0*]
for *i* = 1, …, |*array*| - 1
if *array*[*i*] > *max*
max,*min* = *array*[*i*], *min*
else if *array*[*i*] < *min*
max,*min* = *max*,*array*[*i*]
return *max*, *min*
The computational complexity of this solution is $O(n)$, because there is only one loop with n iterations.