# Multiple Choice Style Informatics

*Jordan Tabov, Emil Kelevedzhiev & Borislav Lazarov*

*Jordan Tabov was an IMO participant and has been a team leader of the Bulgarian IMO team. He graduated from Moscow State University and since then he has been working at the Institute of Mathematics and Informatics of the Bulgarian Academy of Sciences. He is a Vice - President of the Tournament of the Towns and member of the editorial boards of several mathematics magazines and series of books related with mathematics competitions. In 1994 he was awarded the Paul Erdos Award of the WFNMC.*

## I. Introduction.

On 1st November 2001, the tenth edition of the Bulgarian mathematical tournament 'Chernorizets Hrabar' (briefly ChH) was held. As it was the first Bulgarian multiple choice competition and for a long period the unique math competition of this type, ChH has a number of specifics. This paper concerns a feature of ChH which probably distinguishes it from all multiple choice math competitions as far as the authors know and that is the presence of informatics problems.

During the last 20 years informatics has become one of the most developed branches of science closely related to mathematics. This is the main reason for beginning to separate student competitions in informatics. But another reason was the lack of any informatics related problems in mathematics competitions. The result is a complete separation of mathematics and informatics competitions not only in organization but even in student resources: as a rule students who attend math competitions avoid informatics competitions and vice versa. We thought that this was inacceptable 10 years ago and still have the same view now.

So ChH is an attempt to satisfy the demand for designing a math competition which includes enough informatics: at least to the extent of that in plane geometry, combinatorics and arithmetic. From this starting point we have the form of the competition almost uniquely determined – multiple choice problems competition.

But the question that arises is what kind of informatics is suitable for multiple choice problems. The only preliminary requirement for us was that an appropriate problem should allow a solution that can be found without using a computer. Since we started out in an unexplored area we had not any classification in mind. Now, in retrospect, we can say that we focus our attention on three basic directions, listed below.

## II. Examples

(problem numbers refer to the Appendix 2).

**1.** Topics typical for computer programming (Informatics related questions).

   (a) Given a commonly accepted description of an algorithm, compute its result for a given initial data – problems 93-14, 00-11.

   (b) For a given algorithm, find out whether it will stop or will not, or compute how many times a given loop will run – problem 93-24.

   (c) Find out a range of input data for which a given algorithm will complete correctly – problem 93-18.

**2.** Identification of an algorithm – what the algorithm should do? Guess the result, or describe the implemented method, recognizing in it some known algorithm – problems 93-12, 94-26, 96-27, 00-16, 00-17.

   When a result of computation is asked, the difference between 1 and 2 is the following: In case 1, the result can be found by routine computations, while in case 2, one needs to recognize the action by means of applying some theoretical arguments.

**3.** Optimization – find out a minimal or maximal number of operations, which are necessary to compute some values, etc – problem 92-13.

## III. Conclusions.

We have enough indicators to conclude that informatics problems (briefly IP) are relevant to our math competition. They enrich it in several directions.

**1.** The format of an IP differs in standards from the other math problems. Putting IP among usual-looking problems is a kind of shock.

**2.** The IP allows to give another view point to well-known math ideas. Sometimes this new point of view magically transforms Cinderella to a princess.

**3.** The algorithmic style of solving IP encourages students to apply it in pure math problems (see Appendix 2).

Statistics shows that IP are not a stumbling-block for the winners of ChH. As a rule the difficulty of an IP is of moderate level. Some are hard. Some are easy but they are classified by the jury as middle, because of their unusual look. As a direct result of their forms, all of the IP have a high discrimination factor.

It is hard to calculate the exact percentage of IP included in ChH. This is because some problems types are quite fuzzy to be strictly classified, e.g. problems concerning number systems. The proportion of IPs varies between 5% and 12%, but it is not so important how many IPs are contained in a ChH tournament test. Knowing there will be such problems, students (and teachers) include new areas in their training programs. And they like to do this.

## Appendix 1. Features of the ChH tournament

Test of 30 multiple choice questions;

Penalty for each wrong answer;

Time allowed: 90 min;

No calculators or computers permitted;

Number of age levels: 3;

Over 2000 participants;

Where Held: about 10 in all main Bulgarian cities;

Frequency: yearly, every year on November 1.

## Appendix 2. Selected problems and brief solutions.

**92–13.** What is the least possible number of multiplications by which $a^{15}$ can be calculated for a given $a$ (except for multiplications, any other arithmetic operations, such as raising to a power, are not allowed) ?

**A)** 3 **B)** 4 **C)** 5 **D)** 6 **E)** 14

*Solution.* $a^{15}$ can be obtained from $a$ after 5 multiplications in the following manner: First $a^2 = a \cdot a$, and then $a^3 = a \cdot a^2$. We have used two multiplications. The third is $a^5 = a^2 \cdot a^3$ and by 2 more multiplications we obtain $a^{15}$ in the following way: $a^{10} = a^5 \cdot a^5$ and $a^{15} = a^5 \cdot a^{10}$. To prove that we cannot obtain $a^{15}$ by 4 or less multiplications we make the full list of all possible cases

By 1 multiplication, only $a^2$ can be obtained.

By 2 multiplications: $a^3$ and $a^4$.

By 3 multiplications: $a^5$, $a^6$ and $a^8$.

By 4 multiplications we can obtain $a^7$, $a^9$, $a^{10}$, $a^{11}$, $a^{12}$ $a^{16}$.

*Answer.* C).

**93–12.** The values of 128 and 52 are assigned to variables `a` and `b`, respectively. What will the variable `a` hold after a run of the following program part?

While `b>0` do:

`a:=a-b` and if `a<b`, then exchange the values of `a` and `b`.

**A)** 1 **B)** 2 **C)** 4 **D)** 8 **E)** 0.

*Solution.* This part of the program implements Euclid's algorithm for finding out the greatest common divisor of two numbers (in the case, they are assigned to the variables `a` and `b`). When $a = 128$ and $b = 52$, the greatest common divisor is 4.

*Answer.* B).

**93–14.** An algebraic expression can be written in postfix notation (also called reverse polish notation) by putting first the operands and then their operation signs. For example, $a + b$ can be written as $ab+$, and $ab - c\cdot$ is an example of postfix notation of $(a - b) \cdot c$. What is the value of $abc + \cdot d\cdot$, where $a = 2, b = 3, c = 4, d = 5$ ?

**A)** 1 **B)** 7 **C)** 15 **D)** 68 **E)** 70.

*Solution.* We have $abc + \cdot d\cdot = a(7) \cdot d\cdot = (14)d\cdot = 70$.

*Answer.* E).

**93–24.** Variable `a` contains a real number in a floating point form. We assume its initial value is equal to 1. What will happen during the execution of the following loop?

While `a+1` $\neq$ `1` do: `a:=a/2`.

**A)** The execution of the loop will be repeated forever.
**B)** The execution of the loop will never begin.
**C)** The loop will be executed exactly once.
**D)** The loop will be executed finitely many times, but more than once.
**E)** None of the above.

*Solution.* Greater number of times the loop is repeated, closer to the zero variable's value `a` becomes. Due to the peculiarities of the floating point arithmetic, if two numbers have a great difference in their exponents, they have their sum closer (and even equal) to the number with the largest exponent. In our case, after several runs of the loop, the equality `a+1=1` becomes true in the floating point arithmetic, and the loop stops running at that moment.

*Answer.* D).

**94–18.** A computer program adds the integer values 1, 2, ..., $n$ consequently to an integer variable `x`. The numbers are stored in a binary form with 15 digits at most. The initial value of `x` is 0. What is the greatest integer $n$ for which the program can perform a correct computation?

**A)** 126 **B)** 255 **C)** 256 **D)** 1024 **E)** 32768.

*Solution.* The sum of the first $n$ positive integers is $S(n) = n(n+1)/2$. The greatest number that can be stored when 15 bits are used, is $2^{15}-1$. Therefore, we should find such a number $n$, so that $S(n) \leq 2^{15}-1$ and $S(n+1) > 2^{15}-1$, i.e. $n(n+1) < 2^{16} \leq (n+1)(n+2)$. By probing we discover that for $n = 2^8 - 1 = 255$ both the inequalities are satisfied.

*Answer.* B).

**94–26.** Let the variables **a** and **c** have the values $x$ and $y$, respectively. Let the variable **b** have the value 1. What will contain **b** after a run of the following program part:

While **c** $> 0$ do:

if **c** is an odd number then **b:=b.a;**

**c:=** the integer part of **c/2**

and **a:=a.a.**

**A)** $x^y$ **B)** $x^{y-1}$ **C)** $x.y$ **D)** $x-y$ **E)** $x^2$.

*Solution.* If $y = b_n \cdot 2^n + b_{n-1} \cdot 2^{n-1} + \cdots + b_1 \cdot 2^1 + b_0 \cdot 2^0$, $b_k \in \{0;\ 1\}$, then $x^y = x^{b_n \cdot 2^n} \cdot\ \cdots\ \cdot x^{b_1 \cdot 2^1} \cdot x^{b_0 \cdot 2^0}$. The algorithm is storing consequently in the variable **b** the results of the multiplications from right to left in this product, and this is doing only for those terms, for which $b_k = 1$. After each step, the values $x, x^2, x^4, x^8, \ldots$ appear in **a**. The value of **c** (after deleting the remainder) is divisible by 2, and the remainders themselves, when they are divided by 2, are the numbers $b_0, b_1, \ldots$.

*Answer.* A).

**96–27.** Variables `x` and `y` store numbers in a floating point form. What will be the value stored in the variable `x` after a run of the following program part:

```
x = 0; y = 9999;

While x ≠ y repeat

  {

    x = y;

    y = square root of (9999+2*x);

  }
```

**A)** 0 **B)** 1 **C)** 99 **D)** 101 **E)** 9999

*Solution.* The program performs a repeating process $x_{n+1} = \sqrt{9999 + 2x_n}$, $n = 0, 1, 2, 3, \ldots$, $x_0 = 9999$. Obviously $x_n \geq 0$ for $n = 0, 1, 2, 3, \ldots$. The process is convergent because of the condition

$$x_{n+1} - x_n = \sqrt{9999 + 2x_n} - \sqrt{9999 + 2x_{n-1}}$$

$$= \frac{2(x_n - x_{n-1})}{\sqrt{9999 + 2x_n} + \sqrt{9999 + 2x_{n-1}}}$$

Therefore $x_{n+1} - x_n < 0 \iff x_n - x_{n-1} < 0 \iff \ldots \iff x_1 - x_0 < 0$. But $x_1 = \sqrt{3 \cdot 9999} < 9999 = x_0$. From there $x_1 - x_0 < 0$ and $x_{n+1} - x_n < 0$, i.e. $x_{n+1} < x_n \ \forall n \in \mathbf{N}$. Thus the sequence $\{x_n\}$ is decreasing, bounded from below and hence convergent.

The limit satisfies the equation $x^2 = 9999 + 2x$, whose positive root equals 101.

*Answer.* D).

**2000–11.** The disk subdirectory `ChernorizecHrabar` contains the following files `ch1.imm, ch2.imm, ..., ch2000.imm`, all sorted by name. How many consecutively placed files has Albena marked, if the first one is `ch5.imm` and the last one is `ch51.imm`:

**A)** 3 **B)** 13 **C)** 46 **D)** 47 **E)** none of these

*Solution* The marked files are `ch5.imm`, `ch50.imm`, `ch500.imm`, `ch501.imm`, `ch502.imm`,..., `ch509.imm`, `ch51.imm`.

*Answer.* B).

**2000–16.** How many numbers will be printed after the execution of the following program:

```
10 FOR X=-10 TO 10
20 IF X*X-9*X-22=20 THEN PRINT X
30 NEXT X
```

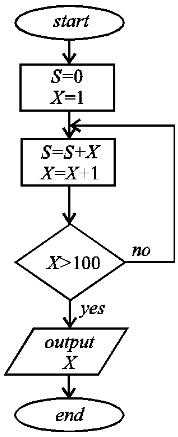
**A)** 0 **B)** 1 **C)** 2 **D)** 3 **E)** 21

*Solution* The program is designed to print out the integer roots of the equation $x^2 - 9x - 42 = 0$ located in the interval $[-10; 10]$; but the equation has no such roots.

*Answer.* A).

**2000–17.** Which number will be printed out after the execution of the algorithm shown on the figure:

**A)** 101
**B)** 4096
**C)** 5050
**D)** 10100
**E)** 1



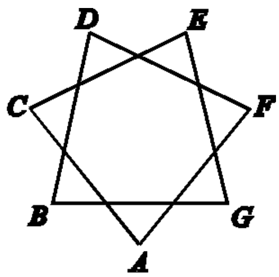

*Solution* The algorithm finds and prints out the sum of all the integers lying between 1 and 100. This sum is 5050.

*Answer.* C).

**Appendix 3.**

**93–6.** What is the sum of the angles $A$, $B$, $C$, $D$, $E$, $F$, $G$ of the star $ABCDEFG$ shown on the figure?

**A)** 180°
**B)** 270°
**C)** 360°
**D)** 540°
**E)** 630°




*Solution:* Using a computer oriented style of thinking:
Imagine you are a turtle, a character well known to the students learning Logo programming language. So, we are using so called 'turtle graphics'. Visiting consequently every vertex along the path $A, C, E, G, B, D, F$, you are turning in an anti-clockwise direction, each time to the some

extent. The sum of all the measures of changing directions gives the answer to the problem.

Let us first consider a special case when the star is regular, i.e. it can be inscribed in a circle. So we have 7 turning points, each yields the same angle of $\alpha = (1/2) \cdot 360° \cdot (3/7)$. The sum is equal to $7 \cdot \alpha = (3/2) \cdot 360° = 540°$.

In the general case, the exact measure in degrees at each turn is unknown, but at the end of the process, we can see that the turtle head directs to the same straight line as it was in the beginning. Hence, the accumulated measure is a multiple of 180°. Due to the reason of continuity between the "regular shape" of the star and its "general shape", we can conclude there is no jump in measure while changing the shape, i.e. the multiple factor is the same as for the "regular shape", hence the answer is 540°.

*Prof Jordan Tabov*
*Institute of Mathematics*
*Acad. G. Bonchev Str. bl. 8*
*1113 Sofia*
*BULGARIA*
*email: jordan@tabov.com*

* * *