Olympiads in Informatics, 2022, Vol. 16, 107–123

DOI: 10.15388/ioi.2022.09


# REPORTS

# Primary School Programming Olympiads in Gomel Region (Belarus)

Michael DOLINSKY
*Faculty of Mathematics and Technologies of Programming, F. Skorina Gomel State University*
*Sovetskaya str., 104, Gomel. 246019. Republic of Belarus*
*e-mail: dolinsky@gsu.by*

**Abstract.** The content of programming contests for students in grades 1–4 of the Gomel region (Belarus) is described in this article. A general idea of the thematic content of the tasks and examples of the tasks of the city Olympiad, which took place on April 8, 2016, is provided. The methodology of teaching and preparing junior schoolchildren for such Olympiads is also briefly described. A serious technical basis is the instrumental system of distance learning developed under the control of the author (DL.GSU.BY).

**Keywords:** programming olympiads, primary school, distance learning instrumental system.

## 1. Introduction

The popular and difficult task of computer science education is introduction of informatics education in primary school (Dagienė *et al.*,2019). One can see many directions of this work: unplugged education (Plugar, 2021; van der Vegt, 2016), gamification process focused on increase in motivation and engagement of the learners (Combéfis *et al.*, 2016), using Scratch (Fagerlund *et al.*, 2020), performing of certain problem solving tasks of controlling an agent or planning its future behavior – in a digital environment: programmable toy, microworld, programming environment (Kabátová *et al.*, 2016), robot programming (Kanemune *et al.*, 2017), learning visual programming, programming and robotics, and programming and electronics (Panskyi *et al.*,2021), using special software (Tsvetkova *et al*., 2021 and Alemany *et al.*, 2016).

Since September 1996, on the basis of secondary school 27 in Gomel, and in September 1999, additionally and on the basis of the distance learning site DL.GSU.BY (hereinafter referred to as DL), work is being carried out on the optional study of computer science and programming for schoolchildren of different ages (Dolinsky, 2016). The key feature of this training is the early start of education – actually from the 1st grade, and in some cases from kindergarten (Dolinsky, 2018). For such students, special programming olympiads are held in order to increase motivation for classes, as well as for the early acquisition of competitive experience. This article offers materials for such Olympiads and a brief description of teaching programming and preparing for such Olympiads for students in grades 1–4. Training includes the sequential study of the necessary information and their consolidation by solving the proposed tasks. Verification of solutions is carried out automatically on the DL.GSU.BY website (Dolinsky, 2017).

## 2. Contents of the Olympiads

Problems for grades 1–4 include three groups of tasks in ascending order of difficulty (each student is invited to solve all these tasks):

**Group 1 of tasks** (10 tasks): includes tasks from the "Introduction to programming" section: three tasks with numbers (Dolinsky, 2019), one each with symbols, strings, line lengths, position of a character in a line and three tasks for using the built-in programming language Pascal of line processing functions: DELETE, COPY, POS, respectively, delete part of a line, copy part of a line and find the position of the first occurrence of one line into another. In tasks 1–10 (each with 5 points), one needs to write a program that works in accordance with the given examples of input and output:

№1

| **Output example:** |
|---|
| 2<br>0<br>1 6 |

№2

| **Input example:** |
|---|
| 18 |
| **Output example:** |
| t=18 C |

| **Input example:** |
|---|
| 24 |
| **Output example:** |
| t=24 C |

№3

| **Input example:** |
|---|
| 17<br>8 1 |
| **Output example:** |
| 17=8+9<br>17=1+16 |

| **Input example:** |
|---|
| 21<br>11 3 |
| **Output example:** |
| 21=11+10<br>21=3+18 |

№4

| **Input example:** |
|---|
| + |
| **Output example:** |
| not + |

| **Input example:** |
|---|
| k |
| **Output example:** |
| not k |

№5

| **Input example:** |
|---|
| Dog<br>bird<br>mouse |
| **Output example:** |
| s2+s1+s3<br>bird+dog+mouse |

| **Input example:** |
|---|
| mountain<br>sea<br>tree |
| **Output example:** |
| s2+s1+s3<br>sea+mountain+tree |

№6

| **Input example:** |
|---|
| mango<br>snow<br>fabiele |
| **Output example:** |
| (snow)=4<br>(fabiele)=7<br>(mango)=5<br>7-5+4=6 |

| **Input example:** |
|---|
| honesty<br>forest<br>sophisticated |
| **Output example:** |
| (forest)=6<br>(sophisticated)=13<br>(honesty)=7<br>13-7+6=12 |

№7

| **Input example:** |
|---|
| Zewitched |
| **Output example:** |
| Bewitched<br>bewitches |

| **Input example:** |
|---|
| Kird |
| **Output example:** |
| Bird<br>Birk |

№8

| Input example: |
|---|
| mou**R**nt**t**ain<br>4<br>pin**M**eap**S**ple |
| **Output example:** |
| mountain<br>pineapple |

| Input example: |
|---|
| gi**p**raf**a**fe<br>3<br>va**k**riab**k**le |
| **Output example:** |
| giraffe<br>variable |

№9

| Input example: |
|---|
| **H**or**se**<br>1 2 |
| **Output example:** |
| h se horse |

| Input example: |
|---|
| mango<br>3 1 |
| **Output example:** |
| man o mango |

№10

| Input example: |
|---|
| S<br>sunset<br>mouse<br>comicas |
| **Output example:** |
| *unset=1<br>mou*e=4<br>omica*=7 |

| Input example: |
|---|
| O<br>mango<br>susurrous<br>honesty |
| **Output example:** |
| mang*=5<br>susurr*us=7<br>h*nesty=2 |

**Group 2 of tasks** (5 tasks): includes tasks from the "One-dimensional array" section: summing elements, counting elements with a certain property, finding the maximum and minimum elements, and finding the number of an element with a given property. The following are examples of such tasks.

**Problem 11 (5 points)**

For several weeks Denis saved money that was given to him for pocket expenses. During the autumn holidays, he and his mother decided to go to the park and take a ride on the merry-go-rounds. Calculate the cost of riding on all the carousels and print "+" if Denis has enough money for all the rides, otherwise print the amount of money that you need to ask your mother to have enough to ride on all the carousels.

| **Input format:** | **Input example:** | **Input example:** |
|---|---|---|
| S – the amount of money Denis has<br>K – number of carousels (K<=12)<br>a[1] – the cost of riding the 1st carousel<br>a[2] – the cost of riding the 2nd carousel<br>…<br>a[K] – the cost of riding the K-th carousel | 1500000<br>5<br>20000<br>15000<br>12000<br>17000<br>25000 | 50000<br>4<br>15000<br>12000<br>25000<br>17000 |
| **Output format:** | **Output example:** | **Output example:** |
| s – the cost of riding all carousels<br>+ / n – the amount that mom should give | 89000<br>+ | 69000<br>19000 |

**Problem 12 (5 points)**

Timofey's dad decided to make a basketball ring for his son. The boy has a basket ball with a diameter of 19 cm. To comply with all the rules of basketball, it is necessary that the diameter of the ring is 26 cm larger than the diameter of the ball. Count the number of rings made correctly.

| **Input format:** | **Input example:** |
|---|---|
| m – number of rings (m<=15)<br>a[1] a[2] … a[m]<br>diameters of rings made by dad | 4<br>45 38 45 41 |
| **Output format:** | **Output example:** |
| K – number of rings suitable for Timofey | 2 |

**Problem 13 (5 points)**

People have always appreciated any kind of thrill-related entertainment. Roller coasters remain the most popular extreme attractions at all times. Calculate the speed of the fastest of the five slides.

| **Input format:** | **Input example:** |
|---|---|
| a[1] – speed Kingda Ka, km/h<br>a[2] – speed Top Thrill Playster<br>a[3] – speed Formula Rossa<br>a[4] – speed Dodonpa<br>a[5] – speed Steel Dragon | 206<br>190<br>240<br>172<br>120 |
| **Output format:** | **Output example:** |
| sp – max speed | 240 |

**Problem 14 (5 points)**

Spring has come in the land of the Moomins. Baby Sniff picked the first spring flowers for N days. Of the flowers he collected, he gave five to his family members, and from the rest he made bouquets. Calculate the number of flowers in the smallest bouquet that baby Sniff can get.

| **Input format:** | **Input example:** |
|---|---|
| N – number of days (N<=15)<br>a[1] – number of flowers in the 1st bouquet<br>a[2] – number of flowers in the 2nd bouquet<br>…<br>a[N] – number of flowers in the N-th bouquet | 5<br>7<br>10<br>14<br>9<br>6 |
| **Output format:** | **Output example:** |
| M – number of flowers in the smallest bouquet | 1 |

**Problem 15 (5 points)**

The Babochkin family is about to go on vacation to the sea. In order to choose the right tour, it is necessary that the day of departure to the sea was the day after the father went on vacation. Determine if there are tour start days that are suitable for the rest of the Babochkin family.

| **Input format:** | **Input example:** | **Input example:** |
|---|---|---|
| m – number of tours (m<=10)<br>s – father's day of vacation<br>a[1] – 1st tour day<br>a[2] – 2nd tour day<br>…<br>a[m] – day of the m-th tour | 10<br>11<br>12<br>14<br>15<br>16<br>17<br>13<br>11<br>12<br>14<br>23 | 7<br>29<br>12<br>1<br>6<br>27<br>5<br>18<br>19 |
| **Output format:** | **Output example:** | **Output example:** |
| + / – | + | – |

**Group 3 of tasks** (5 tasks): designed to differentiate the knowledge of skills and abilities of the most prepared children and includes one simple task on the following topics: two-dimensional array, geometry, strings, research (based on Kangaroo tasks of 2–3 grades), word problem.

**Problem 16 (5 points)**

In one of the hottest countries in the world, they decided to measure the air temperature for M hours for a week. Determine on which day the maximum temperature was reached and at what time.

| **Input format:** | **Input example:** |
|---|---|
| M – number of hours to measure temperature (M<=24)<br>a1 a2 … aM – temperature on Monday<br>b1 b2 … bM – temperature on Tuesday<br>c1 c2 … cM – temperature on Wednesday<br>d1 d2 … dM – temperature on Thursday<br>e1 e2 … eM – temperature on Friday<br>f1 f2 … fM – temperature on Saturday<br>g1 g2 … gM – temperature on Monday | 5<br>28 28 29 29 30<br>30 31 31 31 31<br>32 32 32 32 32<br>30 30 30 31 31<br>32 33 33 33 34<br>31 31 32 33 33<br>29 29 30 31 31 |
| **Output format:** | **Output example:** |
| max – highest temperature<br>n – the hour when the maximum temperature was reached | 34<br>5 |

**Problem 17 (5 points)**

Pavel is going to visit all excursions during L days of rest at sea. Determine the farthest excursion from the place of residence, and how far he will travel in all days if he visits one excursion every day and returns to the hotel.

| **Input format:** | **Input example:** |
|---|---|
| L – number of rest days (L<=14)<br>x1 y1 – coordinates of the 1st excursion<br>x2 y2 – coordinates of the 2nd excursion<br>…<br>xL yL – coordinates of the L-th excursion<br>xe ye – coordinates of the place of residence | 7<br>1 1<br>4 2<br>7 1<br>1 5<br>8 10<br>12 6<br>6 6<br>6 5 |
| **Output format:** | **Output example:** |
| m – distance to the furthest excursion<br>nom – the number of the furthest excursion<br>rast – total distance<br><br>Display real numbers with 1 decimal place. | 6.4<br>1<br>63.2 |

**Problem 18 (5 points)**

N lines are given containing characters '.' and '#'. Print the number of the first line containing the least number of '#' characters.

| Input format: | Input example: |
|---|---|
| N (N<=5)<br>s1<br>s2<br>..<br>sN | 4<br>.##..<br>....<br>######<br>##.#### |
| **Output format:** | **Output example:** |
| k - line number | 2 |

**Problem 19 (5 points)**

The difference between the two numbers is X less than the number to be subtracted and by Y more than the subtracted one. What is it equal to?

| Input format: | Input example: |
|---|---|
| X Y | 3 4 |
| **Output format:** | **Output example:** |
| Z - difference of two numbers | 7 |

**Problem 20 (5 points)**

**Input file: input.txt Output file: output.txt**

While e-books are gaining popularity around the world, in Byteland, everyone loves to read books in the library.

The National Library of Byteland has a variety of shelves, each containing books on a specific topic. The most popular shelving unit is a selection of fiction. The rack has **N** shelves, each of which holds **N** books.

Sometimes readers return a book to the shelf and put it on the wrong shelf. Therefore, every evening the librarian Eleonora Romualdovna puts things in order in the rack. In total, the National Library of Bytelandia contains works of art by **N** different authors, and the library contains $\mathbf{a}_i$ books of the **i**-th author. Eleonora Romualdovna defines the disorder on the shelf by the number **P**, which is equal to the maximum of $\mathbf{p}_i$ values, where $\mathbf{p}_i$ is the disorder on the i-th shelf. The clutter on the **i**-th shelf is calculated as the number of different authors whose works are located on it. Eleonora Romualdovna believes that the rack is in perfect order if the number **P** is minimal.

Arranging books every day, deciding where to put each book, is a very difficult task, so Eleonora Romualdovna asks you to help. To begin with: count how many books she has in the library.

**Input data:**

The first line of the input file contains a single natural number **N** ($1 \le \mathbf{N} \le 100$) – the number of shelves.

The second line contains **N** integers $\mathbf{a_i}$ ($1 \le \mathbf{a_i} \le 10^5$) – the number of books by the **i**-th author.

The numbers in the lines of the input file are separated by single spaces.

**Output data:**

The output file should contain one number – the total number of books in the library.

| *input.txt* | *output.txt* |
|---|---|
| 3<br>1 2 6 | 9 |

---

Systematic and purposeful preparation of regional Olympiads is an important means of developing the Olympiad movement in the region. Regional Olympiads are held in the Gomel region five times a school year: in October-November for school and city at three divisions: grades 1–4, 5–8, 9–11 and in March–April for school, city and regional at three divisions: grades 1–4, 5–7, 8–9. When conducting these Olympiads, Internet technologies and the DL.GSU.BY website are used, which allows not only schoolchildren of the Gomel region to participate in all the Olympiads, but also everyone who wishes. And, it should be noted, there are dozens of such applicants from all regions of Belarus and the city of Minsk.

## 3. Training System

It is important to note that, despite the focus on programming, training is essentially developing in nature and therefore it is very useful both for those who subsequently choose information technology as their professional field, and for everyone who will be engaged in at least some time. Practice also shows that the training is structured in a rather interesting form. All classes are conducted only on a voluntary basis outside the classroom. Another equally important aspect is the differentiated approach. The use of Internet technologies makes it possible to provide individual training along a personal educational trajectory.

The work with first-graders begins with the course “Learning to think”(Dolinsky, 2014). A side effect of the lessons in this course is the growing interest in teaching a wide range of younger students. The main goal is to acquire stable skills in performing basic mental operations. At the moment, the course offers the following basic mental operations (in the amount of 21 pieces):

- Operations on pairs: comparison, ordering, association.
- Operations on sets: union, intersection, subtraction.
- Operations on a set: classification, structuring, generalization.
- Logical operations: negation, conjunction, disjunction, equivalence, implication.

Complex operations: synthesis, memorization, analysis, imagination, analogy, abstraction, positioning.

Further training is consistently conducted within the following sets of tasks "Introduction to programming", "Debugger", "One-dimensional array", "Two-dimensional array", "Geometry", "Strings" (Dolinsky, 2013). Learning in all these sets of tasks is built on the principles of differentiated learning (Dolinsky, 2020). Stem tasks are listed in ascending order of difficulty. For each stem problem, there is a branching tree of leading problems of less complexity. In the end, for each task, training is provided with the presented source code in approximately the following order.

Each studied task after a certain number of tasks is met as a control one. In this case, there is no "Don't know" button. If a student cannot solve such a problem (previously studied) even with the help of his notebook, he is automatically transferred back to learning to this problem. To stimulate more intense thinking activity, as opposed to thoughtlessly pressing the "Don't Know" buttons, most of the lead-in folders are provided with some of the test assignments described above, BEFORE and AFTER learning to solve the problem for which the student pressed the "Don't Know" button.

Folders with tasks for the development of basic mental operations are continuously interwoven into the learning process, both on the basis of graphic images and on the basis of using the studied material as graphic images in the form of tests, algorithms and programs.

For a more complete control of the assimilation of topics, at the end of each of them there are folders with analogy problems. If a student finds it difficult to solve them, then it is necessary to improve learning in general and work with this student, in particular.

Next, there are tasks, for the solution of which the ability to combine the studied methods of solving problems is required.

Finally, each topic ends with a complete set of available Olympiad problems. Since April 2007, programming contests have been held in the Gomel region for students in grades 1–4. First, there are olympiads problems on a given topic, and then sets of olympiads problems on all topics studied up to this topic, inclusive.

## 4. Conclusion

The materials of programming contests for primary school students and briefly presents the methodology for teaching and preparing junior schoolchildren for such Olympiads are considered in this article. The Olympiad for pupils of grades 1–4 of Gomel and the Gomel region, held on April 8, 2016, was attended by 50 students from 10 settlements: Gomel, Rechitsa, Zhlobin, Kalinkovichi, Mozyr, Svetlogorsk, Chechersk (all-Gomel region), Grodno, Lida (Grodno region), Polotsk (Vitebsk region). The winner (4th grade student Kopichenko S.) solved all the proposed problems. Diplomas were awarded to three 1st grade students, two 2nd grade students, five 3rd grade students and eight 4th grade students from Gomel, Zhlobin, Svetlogorsk and Kalinkovichi. The wide geography and high results confirm both the correct choice of a set of tasks and the effectiveness of the proposed distance learning system. The author's solutions to the problems of this Olympiad are attached to the article.

## References


Alemany F.J., Vilahur V.J. (2016). eSeeCode: Creating a Computer Language from Teaching Experiences. *Olympiads in Informatics*, 10, 3–18.

Combéfis S., Beresnevičius G., Dagienė V. (2016). Learning Programming through Games and Contests: Overview, Characterisation and Discussion. *Olympiads in Informatics*, 10, 39–60.

Dagienė V., Jevsikova T., Stupurienė G. (2019) Introducing Informatics in Primary Education: Curriculum and Teachers' Perspectives. In:

Dolinsky M. (2013). An approach to teach introductory-level computer programming. *Olympiads in Informatics*, 7, 14–22.

Dolinsky M. (2014). Technology for the development of thinking of preschool children and primary school children. *Olympiads in Informatics*, 8, 63–68.

Dolinsky M. (2016). Gomel training school for Olympiads in Informatic. *Olympiads in Informatics*, 10, 237– 247.

Dolinsky M. (2017). A New Generation Distance Learning System for Programming and Olympiads in Informatics. *Olympiads in Informatics*, 11, 29–39.

Dolinsky M., Dolinskaya M. (2018). How to Start Teaching Programming at Primary School. *Olympiads in Informatics*, 12, 13–24.

Dolinsky M., Dolinskaya M. (2019). Training in Writing the Simplest Programs From Early Ages, 13, 21–30.

Dolinsky M., Dolinskaya M. (2020). The Technology of Differentiated Instruction in Text Programming in Elementary School Based on the Website `dl.gsu.by`, 14, 37–46.

Fagerlund J., Hakkinen P., Vesisenano M, Viiri, J. (2020). Assessing 4th Grade Students' Computational Thinking through Scratch Programming Projects. *Informatics in Education*, 19(4), 611–640. DOI: 10.15388/infedu.2020.27

Kabátová M., Kalaš I., Tomcsányiová M. (2016). Programming in Slovak Primary Schools. *Olympiads in Informatics*, 10, 125–159.

Kanemune S., Shirai S., Tani S. (2017). Informatics and Programming Education at Primary and Secondary Schools in Japan. *Olympiads in Informatics*, 11, 143–150.

Panskyi T., Rowinska Z. (2021). A Holistic Digital Game-Based Learning Approach to Out-of-School Primary Programming Education. *Informatics in Education*, 20(2), 255–276. DOI: 10.15388/infedu.2021.12

Performance Statistics of Gomel pupils at international and national olympiads in informatics since 1997 up to 2020 (In Russian): `http://dl.gsu.by/olymp/result.asp`

Plugar Z. (2021). Extending Computational Thinking Activities. *Olympiads in Informatics*, 15, 83–89.

Pozdniakov S., Dagienė V. (eds) *Informatics in Schools. New Ideas in School Informatics. ISSEP 2019*. Lecture Notes in Computer Science, vol 11913. Springer, Cham. `https://doi.org/10.1007/978-3-030-33759-9_7`

Tsvetkova M.S., Kiryukhin V.M. (2021). Algorithmic Thinking and New Digital Literacy. *Olympiads in Informatics*, 15, 105–118.

van der Vegt W. (2016). Bridging the Gap Between Bebras and Olympiad; Experiences from the Netherlands. *Olympiads in Informatics*, 10, 223–230.


**M. Dolinsky** is a lecturer in Gomel State University "Fr. Skoryna" from 1993. Since 1999 he is leading developer of the educational site of the University (dl.gsu.by). Since 1997 he is heading preparation of the scholars in Gomel to participate in programming contests and Olympiad in informatics. He was a deputy leader of the team of Belarus for IOI'2006, IOI'2007, IOI'2008 and IOI'2009. His PhD is devoted to the tools for digital system design. His current research is in teaching Computer Science and Mathematics from early age.

## Appendix.

The following are the source codes for solving presented problems in Pascal:

**Problem 1**

```
begin
  writeln(2);
  writeln(0);
  writeln(1,' ',6);
end.
```

**Problem 2**

```
var
  a : longint;
begin
  readln(a);
  writeln('t=',a,' C');
end.
```

**Problem 3**

```
var
  s1,s2,s3,s4,s5 : longint;
begin
  readln(s1);
  readln(s2,s3);
  s4:=s1-s2;
  s5:=s1-s3;
  writeln(s1,'=',s2,'+',s4);
  writeln(s1,'=',s3,'+',s5);
end.
```

**Problem 4**

```
var
  a : char;
begin
  readln(a);
  writeln('not ',a);
end.
```

**Problem 5**

```
var
  a,b,c : string;
begin
  readln(a);
  readln(b);
  readln(c);
  writeln('s2+s1+s3');
  writeln(b,'+',a,'+',c);
end.
```

**Problem 6**

```
var
  s1,s2,s3 : string;
  d1,d2,d3 : longint;
begin
  readln(s1);
  readln(s2);
  readln(s3);
  d1:=length(s1);
  d2:=length(s2);
  d3:=length(s3);
  writeln('(',s2,')=',d2);
  writeln('(',s3,')=',d3);
  writeln('(',s1,')=',d1);
  writeln(d3,'-',d1,
          '+',d2,'=',d3-
d1+d2);
end.
```

**Problem 7**

```
var
  s : string;
  c : char;
begin
  readln(s);
  c:=s[1];
  s[1]:='b';
  writeln(s);
  s[length(s)]:=c;
  writeln(s);
end.
```

**Problem 8**

```
var
  s,p : string;
  k,d : longint;
begin
  readln(s);
  readln(k);
  readln(p);
  d:=length(s);
  delete(s,d-k+1,1);
  delete(s,k,1);
  d:=length(p);
  delete(p,d-k+1,1);
  delete(p,k,1);
  writeln(s);
  writeln(p);
end.
```

**Problem 9**

```
var
  s,p,q   : string;
  d,k1,k2 : longint;
begin
  readln(s);
  readln(k1,k2);
  d:=length(s);
  p:=copy(s,1,k1);
  q:=copy(s,d-k2+1,k2);
  writeln(p,' ',q,' ',s);
end.
```

**Problem 10**

```
var
  s1,s2,s3 : string;
  c        : char;
  p1,p2,p3 : longint;
begin
  readln(c);
  readln(s1);
  readln(s2);
  readln(s3);
  p1:=pos(c,s1);
  p2:=pos(c,s2);
  p3:=pos(c,s3);
  s1[p1]:='*';
  s2[p2]:='*';
  s3[p3]:='*';
  writeln(s1,'=',p1);
  writeln(s2,'=',p2);
  writeln(s3,'=',p3);
end.
```

**Problem 11**

```
var
  a       : array [1..12] of longint;
  s,i,n,p : longint;
begin
  readln(p);
  readln(n);
  for i:=1 to n do readln(a[i]);
  s:=0;
  for i:=1 to n do s:=s+a[i];
  writeln(s);
  if s<=p
    then writeln('+')
    else writeln(s-p);
end.
```

**Problem 12**

```
var
  a     : array [1..15] of longint;
  k,i,n : longint;
begin
  readln(n);
  for i:=1 to n do read(a[i]);
  k:=0;
  for i:=1 to n do
    if a[i]=45 then k:=k+1;
  writeln(k);
end.
```

**Problem 13**

```
var
  a   : array [1..5] of longint;
  k,i : longint;
begin
  for i:=1 to 5 do readln(a[i]);
  k:=a[1];
  for i:=2 to 5 do
    if a[i]>k then k:=a[i];
  writeln(k);
end.
```

**Problem 14**

```
var
  a     : array [1..15] of longint;
  i,k,n : longint;
begin
  readln(n);
  for i:=1 to n do readln(a[i]);
  k:=a[1];
  for i:=2 to n do
    if a[i]<k then k:=a[i];
  writeln(k-5);
end.
```

**Problem 15**

```
var
  a     : array [1..10] of longint;
  k,i,s : longint;
begin
  readln(k);
  readln(s);
  for i:=1 to k do readln(a[i]);
  i:=1;
  while (i<=k) and (a[i]<>s+1) do i:=i+1;
  if i>k
    then writeln('-')
    else writeln('+');
end.
```

**Problem 16**

```
var
  i,j,n,m,a,max : longint;
begin
  readln(m);
  max:=-maxlongint;
  for i:=1 to 7 do
    for j:=1 to M do
      begin
        read(a);
        if a>max
          then begin max:=a; n:=j; end;
      end;
  writeln(max);
  writeln(n);
end.
```

**Problem 17**

```
var
  x,y            : array [1..14] of real;
  xe,ye,m,rast,d : real;
  i,L,nom        : longint;
begin
  readln(L);
  for i:=1 to L do readln(x[i],y[i]);
  readln(xe,ye);
  rast:=0; m:=0;
  for i:=1 to L do
    begin
      d:=sqrt(sqr(xe-x[i])+
              sqr(ye-y[i]));
      rast:=rast+d;
      if d>m
        then begin m:=d; nom:=i; end;
    end;
  writeln(m:0:1);
  writeln(nom);
  writeln(2*rast:0:1);
end.
```

**Problem 18**

```
var
  i,j,k,min,n,nom : longint;
  s               : string;
begin
  min:=maxlongint;
  readln(n);
  for i:=1 to n do
    begin
      readln(s);
      k:=0;
      for j:=1 to length(s) do
        if s[j]='#' then inc(k);
      if k<min
        then begin min:=k; nom:=i; end;
    end;
  writeln(nom);
end.
```

**Problem 19**

```
var
  x,y : longint;
begin
  readln(x,y);
  writeln(x+y);
end.
```

**Problem 20**

```
var
  a     : array [1..100] of longint;
  i,N,K : longint;
begin
  assign(input,'input.txt'); reset(input);
  assign(output,'output.txt'); rewrite(output);
  readln(N);
  for i:=1 to N do read(a[i]);
  k:=0;
  for i:=1 to N do inc(k,a[i]);
  writeln(k);
  close(input); close(output);
end.
```