

*МАТЕМАТИКА И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 2006
MATHEMATICS AND EDUCATION IN MATHEMATICS, 2006
Proceedings of the Thirty Fifth Spring Conference of
the Union of Bulgarian Mathematicians
Borovets, April 5–8, 2006*

**ПРИМЕРНА СИСТЕМА ОТ ОПОРНИ ЗАДАЧИ ПО
ТЕМАТА “АЛГОРИТМИ И ЗАДАЧИ ОТ ТЕОРИЯ
НА ЧИСЛАТА” ЗА ПОДГОТОВКА НА ТАЛАНТЛИВИ
УЧЕНИЦИ ПО ИНФОРМАТИКА**

Коста А. Гъров, Елена Х. Тодорова

Настоящата разработка е посветена на подготовката на талантиливи и изявени ученици за участие в олимпиади и състезания по информатика. Предлагат се “опорни” алгоритми и задачи от теория на числата – задължителни за решаване от учениците при подготовката им за състезания по програмиране. Разработката е предназначена за ръководителите на кръжоци и школи за работа с талантиливи деца в областта на информатиката.

I. Въведение. През май 1989 г. в България се провежда Първата международна олимпиада по информатика (МОИ) за ученици, като в нея участват 16 отбора от 13 страни. На последната 17 по ред МОИ, проведена в Полша, участваха отбори от 76 държави. Прави впечатление, че интересът към олимпиадите по информатика бързо нараства, а обикновено най-добре представилите се ученици на тези състезания след време се реализират като водещи специалисти в областта на компютърните науки, софтуерния и хардуерния бизнес. Прегледът на задачите, давани на различни национални и международни олимпиади показва, че за решаването на тези задачи се изискват сериозни познания в областта на алгоритмите и програмирането, а математическите знания и умения са задължителни и без тях не може да се разчита на успешно представяне. Ето защо крайната цел при подготовката на талантиливите деца за участие в олимпиади по информатика не е формиране на множество от фрагментарни знания и умения, а изграждане на цялостна професионална култура в областта на информатиката и информационните технологии. За постигането на тази цел важна роля играе точното определяне на учебното съдържание, което трябва да се преподава в кръжоците и школите по информатика. Разбира се това учебно съдържание се променя във времето и зависи от различни фактори. Важни за учебното съдържание се оказват развитието и възможностите на компютърната техника, учебните програми за задължителна и извънкласна учебна дейност на учениците, тематиката на задачите давани на различни национални и международни олимпиади и състезания. За пръв път у нас през 1985 г. Асен Рахнев, Коста Гъров и Огнян Гавраилов в “РЪКОВОДСТВО за извънкласна работа по ИНФОРМАТИКА на базата на езика БЕЙСИК”, издадено от МНП, правят опит за определяне на основни теми за подготовка за състезания по програмиране. В по-ново време такъв

опит правят Венета Богданова и Галина Момчева в статията си: “Структуриране на учебното съдържание в извънкласните форми по информатика” публикувана в брой 1 на сп. “Математика и информатика” от 2001 г. В тези материали, както и в почти всички ръководства и помагала за извънкласна работа с изявени и талантливи ученици по информатика, една от първите теми, които се разглеждат, е темата за т.н. “Числови задачи”. В нея обикновено се разглеждат основни алгоритми от теорията на числата и съответните им компютърни програми. Това не е случайно. Известно е, че олимпиадите по информатика са възникнали като аналог на олимпиадите по математика. Освен това има и силна междупредметна връзка между математиката и информатиката. По правило най-добрите състезатели по информатика са и много добри състезатели по математика. Изводът е, че за да си **добър състезател по информатика**, трябва да имаш и **много добра математическа подготовка**.

Настоящата работа е посветена на изграждането на примерна система от **опорни задачи** по темата “Алгоритми и задачи от теория на числата”, която да се използва при подготовката на ученици за успешно участие в олимпиади по информатика. Идеята за изграждане на системи от опорни задачи е дадена в [5]. При тази методика за всяка тема от учебното съдържание се определя **задължителен минимум** от задачи наречени **опорни**, които състезателят трябва да реши по време на своята подготовка за участие в олимпиади по информатика.

II. Примерна система от опорни задачи по темата “Алгоритми и задачи от теория на числата”. Решаването на задачи от горе посочената тема може да започне още в началните класове на обучение – 4-7 клас. Удачно е като **опорна задача № 1** да се разгледат алгоритми и програми за намиране на най-голям общ делител (НОД), познат на учениците от часовете по математика. Даваме **строги определения** на въвежданите математически понятия, за да могат учениците от ранна възраст да **свикват** с формални определения.

Може да се даде следното определение на понятието най-голям общ делител на две естествени числа a и b . Най-голям общ делител на a и b се нарича най-голямото цяло положително число, което едновременно се дели на a и b . За най-голям общ делител на повече от две числа е в сила формулата $\text{НОД}(a_1, a_2, \dots, a_n) = \text{НОД}(\text{НОД}(a_1, a_2, \dots, a_{n-1}), a_n)$.

Най-големият общ делител може да се намери по различни начини. Ще разгледаме два алгоритъма за намиране на $\text{НОД}(a, b)$, носещи името на Евклид.

Алгоритъм на Евклид с изваждане:

Този алгоритъм може да се представи така:

- 1) Ако $a > b$, се изпълнява 4), в противен случай се изпълнява 2);
- 2) Ако $a = b$ следва, че сме намерили $\text{НОД}(a, b)$ – това е стойността на b ;
Ако $a \neq b$ изпълняваме 3);
- 3) Присвояваме $b = b - a$ и се връщаме на 1);
- 4) Присвояваме $a = a - b$ и се връщаме на 1).

Примерният код на задачата изглежда така:

```

long int evclid_razlika(long int a, long int b)
{ while (a!=b)
  { if (a>b) a=a-b;
    else b=b-a;
  }
  return a;}

```

Алгоритъм на Евклид с деление:

Този алгоритъм се извършва, като последователно разделяме целочислено a на b , при което получаваме частно q и остатък r . След това делим b на получения остатък, след което делим r на остатъка r_1 и т.н., докато остатъкът стане равен на нула:

$$\begin{aligned}
 a &= bq + r, & 0 < r < b \\
 b &= r_1q_1 + r_1, & 0 < r_1 < r \\
 r &= r_1q_2 + r_2, & 0 < r_2 < r_1 \\
 r_1 &= r_2q_3 + r_3, & 0 < r_3 < r_2
 \end{aligned}$$

На пръв поглед изглежда, че процесът може да продължи неограничено. Но остатъкът r е по-малък от b , r_1 е по-малък от r и т.н., или $b > r > r_1 > r_2 > \dots$. Тъй като числата $b, r, r_1, \dots$ са цели и неотрицателни, след краен брой деления някое от тях ще стане 0, т.е. за някое n ще имаме $nr_{n-1} = r_n \cdot q_{n+1}$. Тогава $\text{НОД}(a, b) = r_n$.

Словесно този алгоритъм може да се опише по следния начин:

- 1) Ако $a < b$, да се разменят стойностите на a и b и да се изпълни 2) с новите стойности; в противен случай да се изпълни 2).
- 2) Да се разделят a и b до получаване на частно и остатък.
- 3) Ако остатъкът е 0, да се да се изпълни стъпка 5); в противен случай да се изпълни 4).
- 4) Да се смята, че новата стойност на a е b , а новата стойност на b е стойността на остатъка. Да се изпълни 2) с новите стойности на a и b .
- 5) Да се изведе стойността на b , която е търсеният $\text{НОД}(a, b)$, след което се прекратява работата.

Следва програмен фрагмент на C++, реализиращ алгоритъма:

```

long int evclid_delenie(long int a, long int b)
{ long int r;
  { while (b>0)
    { r=b; b=a%b; a=r;}
  }
  return a;}

```

На пръв поглед програмите, реализиращи алгоритъма на Евклид, си приличат много, но при едни и същи a и b , втората програма намира $\text{НОД}(a, b)$ много по бързо. Например при $a = 1234567$ и $b = 2345678$, изважданията в първата програма са 17652, а деленията във втората програма са само 8.

Още в този момент на учениците се обръща внимание върху **ефективността** на алгоритмите. Посочва се, че критериите една програма да бъде “по-добра” най-често са да изисква **по-малко памет** или **по-малко време за изпълнение**. За самостоятелна работа на учениците се дават за решаване следните две задачи:

Задача 1. *Съставете програма за намиране на $\text{НОД}(a, b, c)$.*

Задача 2. Съставете програма за намиране на $\text{НОД}(a_1, a_2, \dots, a_n)$.

Разглеждаме **опорна задача № 2** за намиране на най-малко общо кратно на две цели числа.

Определение. Нека са дадени две цели числа a и b . Минималното естествено число d ($d > 0$) такова, че $a|d$ и $b|d$ се нарича най-малко общо кратно (НОК) на a и b . Отбелязва се с $\text{НОК}(a, b)$ или $[a, b]$. Когато търсим НОК на повече от две числа, е в сила формулата: $\text{НОК}(a_1, a_2, \dots, a_n) = \text{НОК}(\text{НОК}(a_1, a_2, \dots, a_{n-1}), a_n)$. Най-малкото общо кратно може да се намери като се използва зависимостта между него и НОД, а именно:

$$\text{НОК}(a, b) = \frac{a \cdot b}{\text{НОД}(a, b)}.$$

На базата на това определение дефинираме НОК на три и n естествени числа. Първо ще разгледаме намиране на НОК на 3 дадени естествени числа. Нека са дадени числата a, b и c . Тогава $\text{НОК}(a, b, c)$ е най-малкото естествено число, което се дели на a, b и c едновременно.

Доказва се, че:

$$\text{НОК}(a, b, c) = \frac{a \cdot b \cdot c}{\text{НОД}(ab, bc, ca)}$$

и тъй като

$$\text{НОД}(ab, bc, ca) = \text{НОД}(\text{НОД}(ab, bc), ca) = \text{НОД}(b \cdot \text{НОД}(a, c), ca),$$

то

$$\text{НОК}(a, b, c) = \frac{a \cdot b \cdot c}{\text{НОД}(b \cdot \text{НОД}(a, c), ac)}.$$

Следователно трябва да намерим първо $m = \text{НОД}(a, c)$ и после $\text{НОД}(bm, ac)$.

Следва програмен фрагмент за намиране на НОД на n цели числа.

```
const long int n=3;
const long int A[]={12,16,34};
long int nod(long int a, long int b)
{ return (b==0) ? a : nod(b, a%b);}
long int nok(long int a[], long int n)
{ if (n==2)
    return(a[0] * a[1] / (nod(a[0],a[1])));
    else { b=nok(a, n-1);
           return(a[n-1] * b) / (nod(a[n-1],b));}
}
main void()
{cout <<nok(A,n);
 return 0;}
```

За самостоятелна работа се дават за решаване следните:

Задача 1. Да се състави програма за намиране на $\text{НОК}(a, b)$.

Задача 2. Да се състави програма за намиране на НОК на три числа без да се използва НОД.

На базата предишните опорни задачи формулираме **опорна задача № 3** за решаване на **диофантови уравнения** от първа степен.

Определение. Уравнението $ax + by = c$, където a, b, c, x и y са цели числа се нарича **диофантово уравнение от първа степен с две неизвестни**.

За решаването му се използва намирането на НОД. Необходимо и достатъчно условие да е решимо уравнението $ax + by = c$ с цели коефициенти a, b и c в множеството на целите числа е НОД(a, b) да се дели на c . Ако x_0 и y_0 са решения на това уравнение, то всичките му решения са $x = x_0 + \frac{b}{\text{НОД}(a, b)} \cdot k$; $y = y_0 - \frac{a}{\text{НОД}(a, b)} \cdot k$, където k е произволно цяло число. Уравнението $ax + by = c$ има безброй много числени решения, когато е решимо.

Една примерна програмна реализация на алгоритъма за решаване на диофантово уравнение е дадена в [1]. Така по **естествен начин** от опорни задачи 1 и 2 се получи решение на опорна задача 3, която е важна математическа задача. Реализира се **междупредметна връзка** между извънкласната работа по математика и извънкласната работа по информатика.

Поради ограничения в обема на настоящия материал по-нататък няма да даваме строги математически определения на понятията и програмни реализации на алгоритми, а ще посочим литературата, в която са описани такива реализации.

Друга група опорни задачи е свързана с намирането на **прости числа**. Ще споменем, че **първата задача**, давана на състезание по програмиране за ученици в България през 1982 г., е свързана с намирането на разлагането на числата от 2 до 100 на прости множители.

Опорна задача № 4. *Да се състави програма за намиране на всички прости числа от 1 до n , където n се въвежда от клавиатурата.*

На базата на опорна задача № 4, да се състави програма, която проверява хипотезата на Голдбах за естествените числа от 5 до 2000. През 1742 г. Голдбах е изказал следната хипотеза: Всяко четно число по-голямо от 4 е сума на две прости числа. Когато учениците са в по-горните класове на средното училище (след 8 клас) те самостоятелно могат да се опитат да съставят програми проверяващи останалите хипотези на Голдбах, текстовете на които могат да се открият в [2, стр. 40].

Опорна задача № 5. *Да се намерят всички прости делители на дадено естествено число n .*

На базата на опорна задача № 5 може да се състави програма за намиране на първите n **съвършени числа**, т.е. такива които са равни на сумата на всички свои делители, без самото число. Например: $6 = 1 + 2 + 3$ е съвършено число. Тук отново може да се реализира **прекрасна** междупредметна връзка между математиката и информатиката, като се анализира историята на съвършените числа. Известно е, че досега са открити малко на брой съвършени числа и редицата от тях нараства много бързо: **6, 28, 496, 8128, 33550336, 8589869056, ...** Така, когато учениците са в началния стадий на подготовка, те могат да решават поставената задача с директен алгоритъм, т.е. да проверяват за всяко $n = 1, 2, 3, \dots$ дали е съвършено или не. Оказва се, че по този начин учениците не могат да намерят повече от **петото** съвършено число за разумно изчислително време на работа на компютъра. На следващ етап от

подготовката на учениците (когато са в горните училищни класове), решаваме отново поставената задача, като се обръщаме към математическата наука и по-точно към известната теорема на Ойлер: **Ако $2^p - 1$ е просто, то $2^{p-1} \cdot (2^p - 1)$ е съвършено.** Така се достига до проблема за намирането на т.н. **Мерсенови прости числа**, т.е. такива прости числа, които могат да се представят във вида $2^p - 1$, където p е просто число. Сега пък възниква проблем при програмирането, защото дори при малки стойности на p , търсените съвършени числа надхвърлят максималната допустима стойност на **стандартните** целочислени типове на езиките за програмиране. Налага се да се създават допълнителни програмни модули, с помоща на които се реализират операции с “големи” числа. Този пример нагледно показва на учениците връзката между информатиката и математиката.

Друго приложение на опорна задача 5 е намирането на така наречените “приятелски” числа, т.е. такива, че всяко число е равно на делителите на другото. Интересна задача за самостоятелна работа е следната: **За числата 2496, 14288, 15472, x , y е известно, че сумата от делителите на първото е равна на второто; сумата от делителите на второто е равна на третото и т.н. Съставете програма, която да намира x и y .**

Друга група опорни задачи по темата е свързана с преобразуването на числа от една бройна система в друга.

Опорна задача № 6. *Да се състави програма, която превръща цели числа от десетична в p -ична бройна система.*

Опорна задача № 7. *Да се състави програма, която превръща цели числа от p -ична в десетична бройна система.*

Опорна задача № 8. *Да се състави програма, която превръща цели числа от p -ична бройна система в система с основа q .*

Тези опорни задачи трябва да се разглеждат в началния етап на подготовка на учениците, а когато те са в горните класове, се решават същите задачи, но числата са реални (дробни).

Друга група опорни задачи е свързана със задачата за “трите съда” (“Задача на Поасон”).

Опорна задача № 9. *Дадени са три съда с вместимост a, b и c литра, където a, b и c са естествени числа, $c > a > b$, $c \geq a + b - 1$, най-големият от които е пълен, а другите два са празни. Да се отмерят d литра, където d е естествено число и $0 < d \leq a$. Допустими са само такива преливания от един съд в друг, при които става максимално напълване и/или максимално изпразване на някой от съдовете.*

Решения на много от поставените задачи могат да се намерят в [1], [2] и [3], а в [4] могат да се намерят текстове на задачи, давани на различни олимпиади и състезания, за решаването на които се **използват посочените опорни задачи.**

ЛИТЕРАТУРА

[1] А. РАХНЕВ, К. ГЪРОВ, О. ГАВРАИЛОВ. Ръководство за извънкласна работа по информатика на базата на езика “Бейсик”, изд. на МНП, София, 1985.

- [2] П. НАКОВ, П. ДОБРИКОВ. Програмиране ++ Алгоритми, изд. Тор Team Co., София, 2002.
- [3] П. НАКОВ. Основи на компютърните алгоритми, изд. Тор Team Co., София, 1997.
- [4] Е. КЕЛЕВЕДЖИЕВ, З. ДЖЕНКОВА. Алгоритми, програми и задачи, изд. Регалия 6, София, 2004.
- [5] К. ГЪРОВ. Система от опорни задачи при подготовката на талантиливи ученици за участие в олимпиади и състезания по информатика, *Математика и математическо образование*, **33** (2004), 316–321.

Коста Андреев Гъров
бул. Източен 111
4000 Пловдив, България
e-mail: kosgar@pu.acad.bg

Елена Христова Тодорова
ж.к. Тракия бл. 71, вх. Е
4000 Пловдив, България
e-mail: eli_tod@abv.bg

**AN EXEMPLARY SYSTEM OF SUPPORTING PROBLEMS IN THE
THEME “THE THEORY OF NUMBERS” IN THE TRAINING OF
TALENTED STUDENTS IN INFORMATICS**

Kosta Garov, Elena Todorova

The present paper deals with the extracurricular training in Computer Science in the secondary school. A list of the so called “supporting” problems in the theme “The Theory of Numbers” is defined. The problems have to be done by the teacher and the students. The paper is meant for teachers in extracurricular courses, working with talented students in the field of Computer Science.