

## ДОСВІД ПІДГОТОВКИ УЧНІВ ДО ОЛІМПІАД З ІНФОРМАТИКИ

Харченко В. М.

В останні роки серед учнів загальноосвітніх шкіл Чернігівської області намітився спад інтересу до олімпіад з інформатики. З одного боку, на ринку праці відчувається значний попит на програмістів, а, з іншого боку, — серед учнів втрачається зацікавлення до змагань, що розвивають нестандартні підходи до створення алгоритмів, досконале володіння мовою програмування.

Причини спаду зацікавлення криються в особливостях підготовки учнів до олімпіад з інформатики:

1. На відміну від олімпіад з математики, учням 8–11-их класів даються однакові задачі для розв'язання.

2. Для успішного розв'язання олімпіадних задач учні повинні знати теми з математики, які вони ще не вивчали в шкільному курсі математики, а значну частину й не будуть вивчати. Так серед тем, які потрібно освоїти учням, є рекурентні рівняння, числа Фібоначі та Каталана, основні алгоритми на графах, що відносяться до курсу дискретної математики ВНЗ. В останні роки активно використовуються задачі з теорії ігор, математичного програмування, обчислювальної геометрії, елементів лексичного та синтаксичного аналізу, які теж не можна віднести до тем шкільної математики.

3. Учні повинні вільно володіти мовою програмування, навичками налагодження програм на той час, коли вони в школі її ще й не починали вивчати. Вони повинні знати частину алгоритмів, які відносяться до програм вищих навчальних закладів: «жадібні алгоритми», динамічне програмування, використання динамічних структур, методи оптимізації алгоритмів.

Отже, для успішного виступу учнів на олімпіадах учитель повинен розпочати готувати їх з 5–6-их класів, а згідно з думкою М. Долинського — з першого класу. В умовах роботи в тих школах, де такої підготовки не було, як у випадку Ніжинського обласного педагогічного ліцею, доводиться форсовано вивчати основні алгоритми й мову програмування одночасно. Оскільки в ліцей набирають до 10-го класу учнів сільських шкіл області, то доводиться вибирати групу учнів, які мають математичні здібності та відзначаються уважністю й великою працездатністю, умінням самостійно працювати як з традиційною літературою, так і з іншими джерелами, зокрема Інтернет-ресурсами. Відбір групи 3–5 учнів проводиться за результатами вступних іспитів, тестувань, спілкування з учителями математики та фізики. Далі підготовка до олімпіад проводиться на факультативних та індивідуальних заняттях. У ході вивчення основ програмування відбувається й поступове введення в суть та методи розв'язання простих олімпіадних задач.

Досить часто виділяють шість етапів розв'язування олімпіадної задачі з інформатики:

1. **Читання умови.** Особливістю умов олімпіадних задач часто є громіздкість їх формулювань та наявність літературної фабули. Зауважимо, що на початку підготовки учнів лякає громіздкість формулювань, і вони без допомоги вчителя навіть не читають їх. Хоча уважне читання умови дало б змогу їм зрозуміти, що задача не складна.

Щодо фабули задач, то учням доводиться пояснювати, що від неї необхідно формалізуватися й пе-

реформулювати умову у більш звичну, чітку, лаконічнішу. Трапляються випадки, коли фабула винесена у вступну частину умови задачі й тоді її можна прочитати побіжно, шукаючи початку змістовного тексту, який потрібно прочитати уважно. Коли ж фабула не винесена в окремий розділ задачі, то слід налаштувати учнів на уважне прочитання всієї задачі.

Досить часто учні побіжно читають формат вхідних і вихідних даних, хоча це й не правильно. Саме тому під час підготовки до олімпіади, звертаємо увагу на те, що у форматі даних часто приховують ключову умову задачі, без якої задача може бути зовсім іншою.

2. **Побудова математичної моделі.** Основне завдання учня на даному етапі формально і математично строго зрозуміти умову задачі. Водночас бажано розробити формальний алгоритм розв'язування задачі. Досить часто учні не можуть побудувати математичну модель через протиріччя, які виникають після пропуску важливих фраз у ході читання умови задачі. Саме формалізація умови задачі й алгоритму її розв'язування може показати школярам, що потрібно знову перечитати умову.

3. **Побудова загальної схеми розв'язання.** На даному етапі учень, перш за все, визначає, наскільки ефективним є висунутий ним формальний метод й удосконалює його. Якщо на етапі побудови математичної моделі школяреві доводилося визначати, що потрібно зробити, то тепер йому слід визначити, як це потрібно зробити.

4. **Уточнення.** На даному етапі відбувається уточнення розв'язання. Водночас слід ще раз визначитися із структурами й типами даних, які будуть використовуватися, які базові алгоритми будуть виконуватися та в якій послідовності. У даному разі рекомендується навіть написати на папері великі фрагменти алгоритмів для перевірки всіх змінних і підпрограм на потрібність і коректність.

5. **Реалізація.** Даний етап розв'язування задачі передбачає власне написання програми. Зауважимо, що доречно постійно наголошувати про необхідність періодичного зберігання коду програм за допомогою відповідної установки у налагодженні IDE системи програмування.

6. **Тестування й налагоджування.** Після компіляції завершеного варіанту програми, необхідно переконатися в тому, що вона правильно працює. Для цього й використовують тестування. Використовуючи кілька маленьких тестів, які учень повинен придумати сам, слід перевірити правильність написання алгоритму. На жаль, це завдання для багатьох учнів є досить складним. Тому в підготовці до олімпіад потрібно звертати увагу учнів на даний етап.

Уже після перших уроків вивчення програмування на факультативних заняттях розпочинаємо підго-



товку до олімпіад. Для цього, поряд із стандартними задачами, пропонуються задачі, які привчають учнів до специфічної фабули та нестандартних формулювань. Вони дозволяють на практиці розглядати етапи розв'язування олімпіадних задач. Приклад задачі.

**Задача 1. «Гра».** У вільний час однокласники Василь і Петро люблять грати в різні логічні ігри: морський бій, хрестики-нулики, шахи, шашки та в інші ігри. Хлопці вже грали у різні класичні ігри подібного роду, включаючи комп'ютерні. Одного разу їм захотілося зіграти в що-небудь нове, але нічого відповідного знайти не вдалося. Тоді Петро придумав наступну гру «Відгадай»: грають двоє учасників. Перший загадує будь-яке тризначне число, таке, що друга й остання цифри відрізняються одна від одної тільки на одиницю. Далі гравець, що загадав число, перевертає загадане число, міняючи першу й останню цифри місцями, таким чином одержуючи ще одне число. Потім з одержаних двох чисел від максимального числа віднімається мінімальне. Задача другого гравця — угадати за першою цифрою одержаного в результаті віднімання чисел число. Наприклад, якщо Василь загадав число 487, то перестановкою першої і останньої цифри він одержить число 784. Після чого йому доведеться відняти від 784 число 487, внаслідок чого вийде число 297, яке і повинен відгадати Петро за вказаною першою цифрою «2», узятою з цього числа. Петро краще знає математику за Василя, тому практично завжди виграє в іграх такого типу. Але в даному випадку Петро схитрував і спеціально придумав таку гру, в якій він не програє Василю у будь-якому випадку. Річ у тому, що придумана Петром гра має виграшну стратегію, яка полягає в такому: шукане число завжди є тризначним і друга його цифра завжди дорівнює дев'яти, а для отримання значення останньої цифри достатньо відняти від дев'яти першу, тобто в розглянутому раніше випадку остання цифра дорівнює  $9 - 2 = 7$ . Допоможіть Петру ще спростити процес відгадування числа за заданою першою цифрою, написавши відповідну програму.

**Вхідні дані:** у програму вводиться єдина цифра  $K$  ( $0 \leq K \leq 9$ ), яка відповідає першій цифрі одержаного Василем в результаті віднімання від більшого числа меншого.

**Вихідні дані:** потрібно вивести трицифрове число — значення одержаної Василем різниці.

**Приклади**

Уводиться	Виводиться
5	594
9	990

**Розв'язання**

Головне завдання учня під час розв'язування даної задачі — дочитати її уважно до кінця та відкинути літературну фабулу. Учні повинні переформулювати дану умову таким чином: «За введеною першою цифрою числа  $X$ , другою цифрою «9» та правилом знаходження третьої цифри « $9 - X$ » вивести на екран трицифрове число».

Формулювання математичної моделі в цій задачі не потрібно. Учням доречно розпочати із схеми розв'язання задачі. Нехай з клавіатури вводиться цифра  $X$ . У самій задачі пояснено, що друга цифра 9, а третя  $9 - X$ . Тоді код програми запишеться:

```
const c=9;
var x:byte;
begin
```

```
readln(x);
writeln(x,c,c-x);
end.
```

Під час обговорення розв'язання пояснюємо, що константа уведена лише для універсальності програми. Коли замість дев'яти в умові задачі буде якась інша цифра, то достатньо лише замінити нею значення константи.

Після розв'язання задачі про обмін значень двох числових змінних за допомогою трьох змінних пропонуємо розв'язати наступну задачу.

**Задача 2.** Поміняти значення двох цілочисельних змінних без використання допоміжних змінних.

**Розв'язання**

Перше, що приходить на думку учням — це написати  $a:=b; b:=a$ . Проте швидко зрозумівши свою помилку, вони знаходять правильний шлях розв'язання. Їм вдається скласти програму, у якій використовуватимуться операції додавання й віднімання:

```
.....
a:=a+b;
b:=a-b;
a:=a-b;
```

У ході обговорення розв'язання наголошуємо, що дане розв'язання не є ідеальним. Воно має слабкий момент — можливе переповнення (наприклад, при  $a=32700$  і  $b=32701$ ). Звичайно, можна відтермінувати таке переповнення, використовуючи типи Free Pascal `word` чи `int64`. Хоча й для них знайдуться значення, які викликають помилку у виконанні програми.

Після вивчення логічних операцій дану задачу можна розв'язати більш ефективно. Водночас слід пам'ятати, що для операції `xor` справедлива властивість:  $A \text{ xor } (A \text{ xor } B) = B$ . Тоді код програми запишеться:

```
var a,b:Int64;
begin
readln(a,b);
a:=a xor b;
b:=a xor b;
a:=a xor b;
writeln(a,' ',b);
end.
```

Коли поміняти у коді програми тип `Int64` на тип `char`, то програма буде чудово працювати. Зауважимо, що дослідження даної проблеми належить М. Густокашину.

Задачі, які аналогічні до наступної, досить часто викликають труднощі в учнів, у яких є певні труднощі із формалізацією прочитаного.

**Задача 3. «Теленовини».** У місті  $N$  в теленовинах розповіли про надзвичайну популярність серед школярів гуртка з олімпіадного програмування. Керівник гуртка  $X$  сказав, що в середньому його 2,5 учнів за 2,5 дні розв'язують 2,5 задач. Напишіть програму, яка дозволяє визначити скільки задач повністю розв'язують  $V$  учнів гуртка за  $n$  днів.

**Вхідні дані:** у програму вводять два натуральні числа  $V$  і  $n$  ( $1 \leq V, n \leq 360$ ).

**Вихідні дані:** програма повинна вивести одне натуральне число — кількість повністю правильно розв'язаних задач.

**Наприклад**

Уводиться	Виводиться
10 6	24
23 30	276

**Розв'язання**

Перш за все, зауважимо, що структура задачі близька до структури олімпіадних задач. За винятком того, що не вимагається вводити дані із текстового файлу та виводити результат у текстовий файл. Літературна фабула задачі винесена в перше речення, в яке можна не вчитуватися.

При побудові математичної моделі, школярі повинні врахувати, що оскільки 2,5 учні міста  $N$  розв'язують 2,5 задачі за 2,5 дні, то один учень за 2,5 дні розв'язує одну задачу. Тоді  $V$  учнів за  $n$  днів розв'язують  $(V/2,5)n$  задач. У випадку даної задачі слід пам'ятати, що учні не можуть розв'язати, наприклад, 3,5 задачі. У такому разі повністю розв'язаних задач буде 3. Тому слід взяти цілу частину від вказаного числа. Код програми на Free Pascal:

```
var v,n:integer;
begin
  readln(v,n);
  writeln(trunc(v/2.5*n));
end.
```

Дана задача пропонувалася автором на II етапі Всеукраїнської олімпіади в Чернігівській області в 2008–2009 н. р. як проста задача.

Під час вивчення алгоритмів розгалуження поряд із стандартними задачами пропонуємо учням розв'язувати й нестандартні.

**Задача 4.** Дідусь Гасан Хоттабич, після визволення Волькою Костильковим із глиняної амфори, вирішив віддячити свого благодійника. Для цього Гасан Абдурахманович хоче заслати кімнату свого благодійника квадратними персидськими килимами. Кімната Вольки має розміри  $a \times b$ , а довжина сторони килима —  $c$ . Оскільки персидські килими не можна згинати й класти один на один, то допоможіть дідусю визначити, чи можна заслати кімнату чи ні.

**Вхідні дані:** у програму вводять три натуральні числа  $a$ ,  $b$  і  $c$  через прогалину ( $0 < a, b, c \leq 20$ ).

**Вихідні дані:** програма повинна вивести YES, якщо кімнату можна заслати квадратними килимами, або NO, коли зробити це не можна.

**Наприклад**

Уводиться	Виводиться
10 6 2	YES
10 6 3	NO

**Розв'язання**

Передбачається, що учні відкинуть фабулу й сформулюють задачу так: чи можна замостити прямокутник розмірами  $a \times b$  квадратами, довжина сторони якого  $c$ .

При побудові математичної моделі учні повинні врахувати, що якщо цілу кількість килимів не можна помістити по довжині чи ширині кімнати, то повністю її заслати не вдасться. Далі вони повинні сформулювати умову застилання кімнати квадратними килимами: якщо числа  $a$  і  $b$  одночасно діляться націло на  $c$ , то кімнату можна заслати, коли ж хоча б одне із чисел не ділиться націло на  $c$ , то — не можна. Код програми на Free Pascal:

```
var a,b,c:byte;
begin
  readln(a,b,c);
  if(a mod c=0)and(b mod c=0) then writeln('YES')
  else writeln('NO');
end.
```

**Задача 5.** «Подвійні числа». Зайшовши в кімнату до Незнайки, Знайко застав того за дивним заняттям: він записував число, наприклад, 512, потім на па-

пері щось ділив, випишував, переписував і, врешті, біля першого з'явилося інше число: 1. Далі Незнайко записав 513, виконав дії, подібні до попередніх, і поряд з'явилося число 513. Аналогічно були отримані й інші два числа 514 та 257. Зрозумівши ідею записів, Знайко похвалив товариша, але зауважив, що, маючи комп'ютера, не слід користуватися дідівським способом обчислень.

Друзі, допоможіть Незнайці. Напишіть програму, яка б за введеним цілим додатнім числом виводила зв'язане відповідним чином ціле додатне число.

**Вхідні дані:** у програмі міститься одне натуральне число  $N$  ( $512 \leq N \leq 1024$ ).

**Вихідні дані:** програма виводить одне натуральне число  $M$  ( $M > 0$ ), яке отримано за вказаним алгоритмом.

**Наприклад**

Уводиться	Виводиться
512	1
513	513
514	257

**Розв'язання**

Дану задачу наводимо як приклад того, що й назва задачі часто робить підказку учасникам олімпіад. Зокрема, у назві даної задачі недаремно виділено «ДВІ» як підказка. Учні також повинні помітити, що увесь алгоритм дій Незнайки нагадує алгоритм переведення числа у двійкову систему числення. Після отримання двійкового числа, воно перевертається: остання цифра стає першою, передостання — другою і т. д. Утворене так число знову переводиться у десяткову систему числення. Оскільки наперед не відомо, скільки потрібно повторювати дії переведення числа, то потрібно використати цикл repeat. Запишемо код програми:

```
var n,m,i:longint;
begin
  readln(n);
  m:=0;
  repeat
    m:=2*m+(n mod 2);
    n:=n div 2;
    writeln('n=' ,n, ' m=' ,m);
  until n=0;
  writeln(m);
end.
```

Дана задача була запропонована автором на 3 тур Всеукраїнської олімпіади Чернігівської області у 2003–2004 н. р. для учнів 8–9-их класів.

Отже, завдяки постійному використанню нестандартних задач для вивчення навіть перших тем з мови програмування вдається:

1. Сформулювати в учнів уміння працювати з умовами олімпіадних задач.
2. Зацікавити учнів олімпіадним програмуванням.

**Література**

1. Брудно А.Л., Каплан Л.И. Московские олимпиады по программированию. — М.: Наука, 1990. — 208 с.
2. Долинский М.С. Решение сложных и олимпиадных задач по программированию: Учебное пособие. — СПб.: Питер, 2006. — 366 с.
3. Караванова Т.П. Информатика: основы алгоритмизации та програмування: 777 задач з рекомендаціями та прикладами: Навч. посіб. для 8-9 кл. із поглибл. вивч. інф-ки. — К.: Генеза, 2006. — 286 с.
4. Кормен Т.Х., Лейзерсон И.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ, 2-е изд. — М.: Изд-ский дом «Вильямс», 2005. — 1296 с.
5. Порублев И.Н., Ставровский А.Б. Алгоритмы и программы. Решение олимпиадных задач. — М.: Изд-ский дом «Вильямс», 2007. — 480 с.