ПОЧИНАЄМО ГОТУВАТИСЯ ДО ОЛІМПІАДИ

Єгорова М.Е.

Сучасний світ — динамічний. Наше життя стало безперервним процесом адаптації, і саме від адаптаційного потенціалу особистості значною мірою залежать її успіх і можливості самореалізації. Нині створюється нова школа, де учень повноцінно живе, проектує своє майбутнє, свій шлях, враховуючи власні можливості, ставлячи перед собою завдання самовдосконалення, самовиховання, самоосвіти. Водночас, особливого значення набуває креативність особистості, її здатність до творчого нестандартного мислення, уміння ефективно розв'язувати складні проблеми власної життєдіяльності.

Саме тому перед учителем постає глобальна задача — своєчасно підготувати учнів до нових умов життя і професійної діяльності у високоавтоматизованому інформаційному суспільстві, навчити їх самостійно діяти в ньому, ефективно використовуючи всі його можливості.

Особливо це важливо під час підготовки учнів до участі в олімпіадах і різноманітних конкурсах. Необхідно допомогти кожному учневі сформувати свою власну траєкторію як навчання, так і самонавчання, самовдосконалення.

Учитель повинен:

- вчити учнів отримувати позитивні результати і створювати для цього всі необхідні умови;
- встановлювати безпосередній зв'язок між результатами праці й оцінкою учнів;
- вивчати потреби учнів, щоб знати, які винагороди мають для них цінності.

Головний принцип підготовки олімпіадників — це системність. Ефективна система підготовки учнів до олімпіад повинна містити:

- навчання на високому рівні складності;
- навчання за провідної ролі теоретичних знань;
- вивчення програмного матеріалу прискореним темпом;
- усвідомлення учнями процесу навчання;
- самостійну роботу й роботу в команді;
- навчання із зацікавленістю і захопленням.

Індивідуальні форми навчання повинні поєднуватися з корпоративними. Такі форми передбачають формування динамічних різновікових груп, на допомогу яким приходять учителі, випускники, фахівці. Учителеві треба вдало поєднувати навчання і роботу учнів з різноманітними інформаційними ресурсами: книжки, посібники, електронні колекції задач, спеціалізовані сайти олімпіадної інформатики.

Але які б форми роботи з талановитими учнями ви не обирали у процесі підготовки до олімпіад, головним завжди залишається самостійна напружена праця учня і його самовдосконалення. Тому важливішою складовою роботи вчителя є визначення індивідуальної траєкторії навчання для кожного учня й організація його самопідготовки. Під час організації самопідготовки учня, роботи з електронними колекціями задач, розв'язуванням і їх перевіркою чудовими помічниками можуть стати спеціалізовані сайти олімпіадної інформатики. Але як навчити учня використовувати свої знання під час розв'язування задач, це завжди залишається важливішою проблемою для вчителя. Для розв'язання її вчитель разом з учнем проходить три етапи до олімпійського успіху.

Накопичення і, ще раз, накопичення основних знань, стандартних алгоритмів, первинних методів розв'язування задач. На цьому етапі учні невпевнені, бояться відірватися від учителя, ще не мають власного почерку розв'язування задач.

Накопичення і, ще раз, накопичення вмінь і навичок. Застосування своїх знань, створення власного підходу до розв'язування задач, розвиток самостійності. На цьому етапі учень впевнений у власних силах, він має велику базу знань, умінь і навичок.

Самовдосконалення, самоосвіта, результати й успіх. На цьому етапі учень самостійно працює з різними інформаційними ресурсами, його постійними помічниками стають різноманітні Інтернет-ресурси, колекції олімпіадних задач, сайти олімпіадної інформатики.

Кожен етап потребує професійної напруженої щоденної праці як учителя, так і учня. Але на початку роботи складніше всього навчити учня розв'язувати задачі. Наприклад, під час вивчення рядкових величин учні засвоюють процедури і функції опрацювання цих величин, а далі?

Ми пропонуємо наші наробки, які використовуємо для вивчення цієї теми. Усе розмаїття задач теми можна поділити на деякі типи і визначити способи їх розв'язування.

Розглянемо три найпоширеніших типи таких задач.

І тип задач, коли потрібно розглянути усі символи тексту, тобто пройти від першого символу до останнього з кроком 1.

Для розв'язування таких задач нам знадобиться оператор повторення з параметром **(FOR)**.

Алгоритм: проходимо від першого до останнього символу (або навпаки), якщо умова виконується, то виконати потрібні дії.

Фрагмент розв'язку має вигляд:

```
...
Write('Ввести текст - '); readln(s);
...
For i:=1 to length(s) do begin ...
     (Або For i:=length(s) downto 1)
      If  умова then оператори;
       ...
      End;
```

ІІ тип задач. Це задачі, коли нам потрібно «пройти» від першого до останнього символу, але з різним кроком або кроком, який не дорівнює 1 (−1).

Алгоритм. Починаємо з першого символу. Поки не дійдемо до останнього, якщо умова пошуку істинна, то виконуємо низку дій і робимо крок1, інакше інші дії та інший крок.

Фрагмент розв'язку має вигляд:

```
...
Write('Введіть текст - '); readln(s);
...
 i:=1;
while i<=length(s) do begin ...
      If  умова then begin оператори; крок 1; end
                 else begin оператори; крок 2; end;
       End;
```

III тип задач. Існують задачі, коли потрібні не всі символи, а тільки фрагменти тексту.

Алгоритм. Поки в тексті існує заданий фрагмент, виконуємо дії:

дії за умовою;

видалення даного фрагменту із тексту для того, щоб знайти наступний.

Фрагмент розв'язку має вигляд:

```
...
Write('Ввести текст - '); readln(s);
Write('Ввести фрагмент тексту - '); readln(s1);
...
While pos(s1,s)>0 do begin {поки існує позиція фрагменту в тексті}
    Оператори; {виконуємо необхідні дії}
     Delete(s,pos(s1,s),length(s1));{видаляємо знайдений фрагмент}
        ...
       End;
```

Розглянемо декілька задач

Задача 1

Задані символи $s_1,s_2,s_3,s_4,\ldots$ Відомо, що символ s1 відмінний від пропуску і що серед $s_2,s_3,\ldots$ є хоча б один пропуск. Розглядаються $s_1,s_2,\ldots s_n$ — символи, які попередні першому пропуску, n наперед невідомо. Одержати послідовність символів $s_1,s_2,\ldots s_n$, замінивши всі малі символи однойменними великими.

Це перший тип задач, тобто нам потрібний оператор повторення з параметрами і функція **UpCase**.

Для латинських символів сміливо застосуємо функцію **UpCase**, для російських символів потрібно звертатися до кодової таблиці і визначати залежність між кодами великих і маленьких літер. Далі, знаючи цю залежність, будемо переходити від коду прописної літери до коду відповідної великої і виводити символ на екран за цим кодом.

```
Program t1;
var s: string;
n, i, a, b: integer; {a, b- це різниця між російськими літерами великими та маленькими}
begin  write('введіть символи'); readln(s);
 a:= 32;{наприклад}
 b:= 80;
 n:=pos(' ',s); {Визначаємо позицію першого пропуску}
 for i:=1 to n-1 do { Перебираємо усі символи від 1-го до цього
                 пропуску}
  case s[i] of      {Вибираємо }
  'a'..'z':s[i]:=upcase(s[i]); {латинські переводимо у верхній регістр}
  'a'..'п':s[i]:=chr(ord(s[i])- a)
  else s[i]:=chr(ord(s[i])-b);
  end;
writeln(s);
end.
```

Задача 2

Перевірити, скільки і яких у заданому тексті зустрічається слів паліндромів. Слова розділені пропуском.

Ідея розв'язку:

- необхідно пройтися від першого до останнього символу тексту і знайти слово S1. Слово S1 одержуємо, додаючи по літерах літери зі слова S, поки літера не є пропуском;
- визначити, чи є S1 паліндромом. Для цього знайдене слово S1 переписуємо навпаки в S2 і порівнюємо S2 з даним.

```
Program t2;
var s,s1,s2: string;
  i,k,j : byte;
begin  write('Введіть текст '); readln(s); k:=0;  s1:='';
   for i:=1 to length(s) do begin
    if s[i]<>' ' then s1:=s1+s[i] {одержуємо слово, додаємо до слова s1 символи із тексту S}
            else begin   s2:='';
            for j:= length(s1) downto 1 do s2:=s2+s1[j]; {перевертаємо  його}
            if s2=s1 then begin inc(k); writeln(s2); end; { порівнюємо }
                    s1:=''; s2:=''; {усе видаляємо і починаємо спочатку}
                end;
    end;
   if k<> 0 then writeln('в рядку ',s,' ',k,' паліндромів')
       else writeln('в тексті паліндромів немає');
   end.
```

Задача 3

Задано рядок, у якому є текст у круглих дужках. Видалити його разом з дужками.

Цю задачу можна розв'язати різними способами. Ось один з них.

Визначити, чи є дужка «(» в тексті. Якщо є, то починаючи з наступної позиції, де вона розташована, підраховуємо скільки символів до наступного символу «)». Вирізаємо з тексту цей фрагмент.

```
Program t3;
var s: string;
  i,n: byte;
begin readln(s);
   n:=pos('(',s);
   if n=0 then writeln('no!!')
       else begin
          i:=n;
          while s[i]<>')' do inc(i);
            delete(s,n,i-n+1);
            writeln(s);
          end;
end.
```

Можна спростити розв'язання, якщо відразу шукати позиції двох дужок:

```
Program t3_2;
var s: string;
  n,n1: byte;
begin readln(s);
   n:=pos('(',s);   n1:=pos(')');
   if n=0 then writeln('no!!')
       else  delete(s, n , n1-n);
            writeln(s);
          end;
end.
```

Задача 4

Задано рядок, у якому є фрагмент, виділений символами «:». Видалити цей фрагмент разом із символами «:».

Визначити, чи є символ «:» у тексті. Якщо є, то видаляємо символи, починаючи з наступної позиції до наступного символу «:».

```
Program t4;
var s: string;
  i,n: byte;
begin readln(s);
    n:=pos(':',s);
    if n=0 then writeln('no!!')
        else begin
           i:=n;
            while s[i]<>':' do begin inc(i);
                              delete(s,n,1);
                               end;
              writeln(s);
           end;
end.
```

Можна спростити розв'язання, якщо відразу шукати позиції двох символів «:» і замінити їх іншим символом, наприклад, «*»:

```
Program t4_2;
var s: string;
  n,n1: byte;
begin readln(s);
    n:=pos(':',s);   s[n]:='*';
    n1:=pos(':');  s[n1]:='*';
    if n=0 then writeln('no!!')
        else  delete(s, n , n1-n);
              writeln(s);
           end;
end.
```

Задача 5

Заданий рядок символів. Перетворити послідовність символів так: видалити кожний символ «*» і повторити всі інші.

Це задача II типу.

```
program task4;
var i: word;
  s: string;
begin    write('input text');   readln(s);   i:=1;
      while i<=length(s) do begin
         if s[i]='*' then  delete(s,i,1)
                 else begin  insert(s[i],s,i+1); i:=i+2;
                    end;
      end;
      write(s);
          readln;
end.
```

Задача 6

У заданому тексті видалити зайві пропуски між словами, залишивши по одному.

Цю задачу можна розв'язувати різними способами, але ми віднесемо цю задачу до III типу і скористаємося III способом розв'язування задач.

Поки в тексті є фрагмент з двох пропусків, будемо видаляти з нього один пропуск.

```
program task6;
var i, n: word;
  s: string;
begin
          writeln('input text');   readln(s);
while pos('  ',s)>0 do delete(s,pos('  ',s),1);
 {поки існує фрагмент з двох пропусків, видаляємо
один пропуск}
        writeln(s);
      readln;
end.
```

Учням можна запропонувати такі задачі:

1. Визначити, чи задовольняє даний текст такій умові: текст починається з деякої ненульової цифри, за якою слідують тільки літери, і їх кількість дорівнює числовому значенню цієї цифри.

2. Текст заданий так: перший символ-цифра, яка вказує довжину першого слова, за першим словом — знов цифра, яка вказує довжину другого слова (довжина кожного слова не більше 9) і т. д. Знайти і вивести на екран *K*-е слово з тексту.

3. Визначити, чи задовольняє текст таким умовам: текст містить (крім літер) тільки одну цифру, причому її числове значення дорівнює довжині тексту.

4. У заданому тексті вставити між словами замість одного пропуску кому і пропуск.

5. Задано текст. Після кожного слова записати в дужках його довжину.

6. Скільки разів у тексті зустрічається задане слово (слова розділені пропусками).

7. Задано текст. Видалити з нього всі літери ***b***, безпосередньо перед якими знаходиться літера ***c***.

8. Перетворити заданий текст, видаливши всі знаки «+», безпосередньо за якими слідує цифра.

9. Замінити в тексті всі слова **child** на слова **children**.

10. Визначити, скільки разів у тексті зустрічається дана літера.

11. У заданому тексті замінити всі слова A1 на слова A2 (довжини слів не співпадають).

12. Визначити кількість слів у тексті.

13. Визначити кількість у тексті подвоєних літер ***a***. Якщо такої пари в тексті немає, відповіддю буде 0. Послідовність **aaa** чи **aaaa** подвоєним **a** не враховувати.

14. Задано текст. Поділити його на рядки довжиною *K*. У разі роздільника в тексті використовувати символ «%».

15. Скласти програму визначення кількості слів і речень в тексті, якщо слова розділяються пропуском, а речення — крапкою.

16. Перевірити, чи є задане слово паліндромом (наприклад, такі як ДІД, БІБ, НАГАН)

17. Вивести номер позиції заданого тексту, у якому в п'ятий раз зустрічається літера «*W*». Якщо такої позиції немає, видати повідомлення про це.

18. Скільки літер «о» стоять в заданому тексті на парних місцях.

Інтернет-ресурси

1. http://www.e-olimp.com/problems — Інтернет-портал організаційно-методичного забезпечення дистанційних олімпіад з програмування Житомирського державного університету імені Івана Франка.
2. http://www.olymp.vinnica.ua — Вінницький центр підтримки та проведення олімпіад школярів.
3. http://acmp.ru/ — сайт «Школа программиста» Красноярського краєвого Палацу піонерів та школярів.
4. http://algolist.manual.ru/ — сайт «Алгоритмы, методы, исходники».
5. http://g6prog.narod.ru/ — сайт «Разбор олимпиадных задач по информатике от Михаила Густокашина» з лекціями з олімпіадної інформатики.
6. http://informatics.mccme.ru/moodle/ — сайт дистанційної підготовки з інформатики.
7. http://www.intuit.ru/catalog/algorithms/algorithms/ — сайт Інтернет-університету інформаційних технологій.
8. http://ips.ifmo.ru — сайт Російської Інтернет-школи інформатики та програмування.