

# Основные направления подготовки учащихся к олимпиаде по информатике. Решение олимпиадных задач



*Калашникова Е.В.,  
МБОУ «Чернянская средняя  
школа №1 с УИОП»*

Для эффективной подготовки школьников к олимпиадам по программированию необходимы четыре условия:

- 1) достаточный уровень логического мышления;
- 2) трудолюбие и целеустремленность школьника;
- 3) достаточное время для подготовки к олимпиаде;
- 4) квалифицированное руководство подготовкой.



# *Первый уровень* подготовки (муниципальный)

- хорошее знание инструмента – языка программирования;
- умение применять технологию проектирования программ;
- владение технологией отладки программ;
- знание простейших алгоритмов:
  - поиск элемента в массиве;
  - сортировка массива;
  - вычисление площади многоугольника;
  - определение положения точки относительно прямой;
  - генерация перестановок.

# Методика обучения

1. Изучение операторов языка программирования
2. Решение задач

Подбор задач должен быть таким, чтобы для их решения требовалось применение различных стандартных приемов или алгоритмов

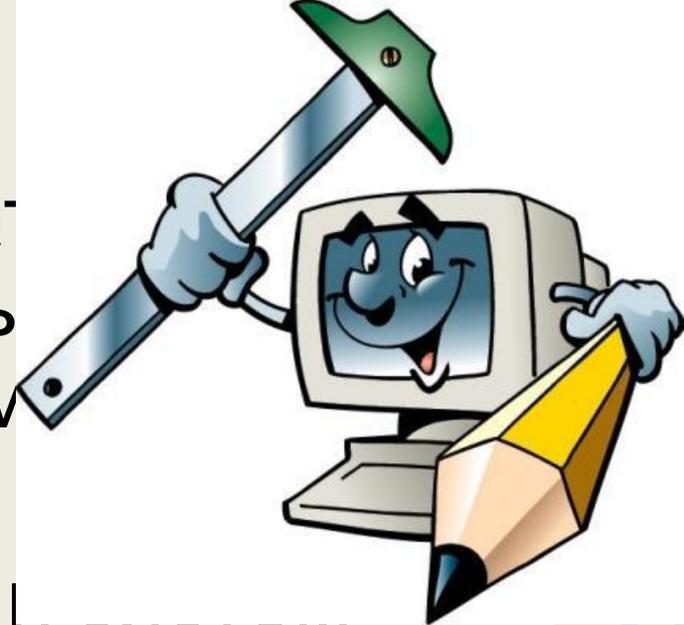


# Этапы решения задачи:

- моделирование задачи, ее формализация и подбор контрольного примера;
- разработка алгоритма решения задачи;
  - рекомендуемой формой записи алгоритмов являются блок-схемы. В блок-схемах сочетаются наглядность и краткость записи.
  - Основными принципами, которые следует соблюдать при разработке алгоритма, являются:
    - проектирование "сверху-вниз" (поэтапная детализация); последовательность построения алгоритма "от главного к второстепенному";
    - применение типовых алгоритмических структур.
  - На начальных этапах обучения программированию разработка блок-схем должна быть обязательным шагом проектирования программы. Рисование блок-схем приучает школьников мыслить типовыми алгоритмическими структурами.

- Разные задачи можно решать с использованием разных языков программирования и систем программирования.

- Список допустимых языков программирования устанавливается предметно-методической комиссией по информатике соответствующего этапа до начала проведения олимпиады с учетом рекомендаций центральной предметно-методической комиссией по информатике.



# Основная группа

Язык	Транслятор	Среда программирования	Операционная система
C/C++	GNU C/C++ 4.2	<u>CodeBlocks 10.05,</u> <u>Eclipse CDT</u>	Любая
C/C++	Microsoft <u>Visual C++ 2005</u>	<u>Встроенная</u>	MS <u>Windows</u>
<u>Object Pascal</u>	<u>Free Pascal 2.4.0</u>	Lazarus 0.9.30	Любая
<u>Object Pascal</u>	Borland/Embarcadero Delphi 7.0	<u>Встроенная</u>	MS <u>Windows</u>

# Дополнительная группа

Язык	Транслятор	Среда программирования	Операционная система
<u>Borland C/C++</u>	<u>Borland C++ 3.1</u>	<u>Встроенная</u>	MS <u>Windows</u>
C#	Microsoft <u>Visual C# 2005</u>	<u>Встроенная</u>	MS <u>Windows</u>
C#	<u>Mono 2.0</u>	<u>MonoDevelop</u>	Любая
<u>Borland Pascal</u>	<u>Borland Pascal 7.0</u>	<u>Встроенная</u>	MS <u>Windows</u>
<u>Visual Basic</u>	Microsoft <u>Visual Basic 2005</u>	<u>Встроенная</u>	MS <u>Windows</u>
<u>Python 3</u>	Python3.1	IDLE или <u>Wing IDE</u>	Любая
<u>Java</u>	<u>Oracle JDK 1.6</u>	<u>Eclipse JDT</u>	Любая

Входной файл – `sort.in`  
Выходной файл – `sort.out`

Время тестирования – 2 секунды  
Максимальная оценка – 100 баллов

Дан текст

алфавита, имеющих значение от буквы **a** до **z**. Необходимо упорядочить эти буквы по алфавиту (должно быть выполнено условие  $a \leq b \leq c \leq d \leq \dots \leq z$ ).

## Входной файл

В первой строке файла содержится число  $N$  ( $1 \leq N \leq 500000$ ). Во второй строке  $N$  случайных строчных букв латинского алфавита, имеющих значение от буквы **a** до **z** без разделителей.

## Выходной файл

В первой строке файла содержится число  $N$  ( $1 \leq N \leq 500000$ ). Во второй строке  $N$  этих букв, упорядоченных по алфавиту без разделителей.

## Пример

Sort.in	Sort.out
<b>13</b> azbzcdddcdbzd	<b>13</b> abbccdddddzzz

```
program z_1;
var f1,f2:text;
    c,i:char;
    cn: array['a'..'z'] of longint;
    j,N:longint;
```

```
begin
```

```
Assign(f1,'sort.in');
```

```
{Связываем переменную f1 с файлом}
```

```
reset(f1);
```

```
{Открываем файл для чтения}
```

```
readln(f1,N);
```

```
{Считываем количество элементов}
```

```
for i:='a' to 'z' do
```

```
cn[i]:=0;
```

```
for j:=1 to N do
```

```
begin
```

```
read(f1,c);
```

```
{Считываем символ}
```

```
Inc(cn[c]);
```

```
{Увеличиваем на 1 количество этих элементов}
```

```
end;
```

```
close(f1);
```

```
{Закрываем файл f1}
```

```
Assign(f2,'sort.out');
```

```
{Связываем переменную f2 с файлом}
```

```
rewrite(f2);
```

```
{Открываем файл для записи}
```

```
writeln(f2,N);
```

```
for i:='a' to 'z' do
```

```
for j:=1 to cn[i] do write(f2,i);
```

```
{Вывод в лексикографическом
```

```
порядке}
```

```
close(f2);
```

```
{Закрываем файл f2}
```

```
End
```

**Text** используется

text

**Char** - описывает

char

**Longint** – целочисленный тип; [-2 млрд. .. 2 млрд.]

longint

значением может быть любой символ

## Задача 1. Золотой песок

7-8 класс

Входной файл – gold.in

Время тестирования – 1 секунда.

Выходной файл – gold.out

Максимальная оценка – 100 баллов.

Имеется три вида золотого песка различной стоимости. Один килограмм песка первого вида стоит  $A_1$  рублей, второго вида -  $A_2$  рублей, а третьего -  $A_3$  рублей. Требуется заполнить этими видами песка три емкости, вмещающими  $V_1$ ,  $V_2$  и  $V_3$  килограммов золотого песка так, чтобы выполнялись условия:

- в одной емкости может содержаться только один вид золотого песка;
- емкости заполняются полностью;
- суммарная стоимость золотого песка в емкостях максимальна.

Разработать программу, вычисляющую максимальную стоимость золотого песка в емкостях.

### Входной файл

Входной файл содержит 6 натуральных чисел  $A_1$ ,  $A_2$ ,  $A_3$ ,  $V_1$ ,  $V_2$ ,  $V_3$ , записанных в одной строке через пробел. Все числа не превосходят 200.

### Выходной файл

Выходной файл *должен* содержать единственное целое число – стоимость золотого песка в трех емкостях в рублях.

Пример:

gold.in	gold.out
1 2 3 3 2 1	14

```
var A1,A2,A3,B1,B2,B3:longint;  
    S:longint;  
    F1,F2:text;  
Procedure Obmen(var a,b: longint);  
var r:longint;  
begin  
    if b>a then begin  
        r:=a; a:=b; b:=r;  
    end;  
end;
```

```
begin  
    Assign(F1,'Gold.in');  
    reset(F1);  
    readln(F1,A1,A2,A3,B1,B2,B3);  
    Close(F1);
```

```
    Obmen(A1,A2); Obmen(A2,A3); Obmen(A1,A2);  
    Obmen(B1,B2); Obmen(B2,B3); Obmen(B1,B2);  
    S:=A1*B1+A2*B2+A3*B3;
```

```
    Assign(F2,'Gold.out');  
    rewrite(F2);  
    writeln(F2,S);  
    Close(F2);  
end.
```



**Желаю**  
**успехов!**  
**Дорогу осилит**