

ЭЛЕМЕНТЫ ТЕОРИИ АЛГОРИТМОВ И ПРОГРАММИРОВАНИЕ



ЗАДАНИЕ 6 (БАЗОВЫЙ УРОВЕНЬ, ВРЕМЯ – 4 МИН)

Тема: Анализ программы с циклом.

Что проверяется:

Знание основных конструкций языка программирования, понятия переменной, оператора присваивания.

1.7.2. Основные конструкции языка программирования. Система программирования.

1.1.4. Читать и отлаживать программы на языке программирования.

Что нужно знать:

- основные конструкции языка программирования: объявление переменных, оператор присваивания, оператор вывода, циклы
- уметь выполнять ручную прокрутку программы
- уметь выделять переменную цикла, от изменения которой зависит количество шагов цикла
- уметь определять количество шагов цикла
- уметь определять переменную, которая выводится на экран
- формулу для вычисления n -ого элемента арифметической прогрессии:
- формулу для вычисления суммы первых n членов арифметической прогрессии: $a_n = a_1 + d(n-1)$

$$S_n = \sum_{i=1}^n a_i = a_1 + a_2 + \dots + a_n = \frac{a_1 + a_n}{2} \cdot n = \frac{2a_1 + d(n-1)}{2} \cdot n$$

где a_i – i -ый элемент последовательности, d – шаг (разность) последовательности

СПОСОБЫ РЕШЕНИЯ

1. Теоретическое
2. Перебор с помощью программы с ручным вводом
3. Перебор с помощью программы
4. С помощью электронных таблиц

6

Определите, при каком наименьшем введённом значении переменной s программа выведет число 64. Для Вашего удобства программа представлена на четырёх языках программирования.

| Паскаль | Python | Алгоритмический язык |
|---|---|--|
| <pre>var s, n: integer; begin readln (s); n := 1; while s < 51 do begin s := s + 5; n := n * 2 end; writeln(n) end.</pre> | <pre>s = int(input()) n = 1 while s < 51: s = s + 5 n = n * 2 print(n)</pre> | <pre>алг нач цел n, s ввод s n := 1 нц пока s < 51 s := s + 5 n := n * 2 кц вывод n кон</pre> |
| C++ <pre>#include <iostream> using namespace std; int main() { int s, n; cin >> s; n = 1 ; while (s < 51) { s = s + 5; n = n * 2; } cout << n << endl; return 0; }</pre> | | |

Ответ: _____.

ПЕРЕБОР С ПОМОЩЬЮ ПРОГРАММЫ С РУЧНЫМ ВВОДОМ

- можно набрать программу и запускать её много раз, вводя различные значения s
- ввод 1 – получаем 1024; ввод 100 – получаем 1; вывод – при увеличении s значение, которое выводит программа, уменьшается
- двоичный поиск: берём середину отрезка $[1;100]$, для числа 50 получаем 2, то есть искомое значение меньше 50
- взяв 25, получаем 64
- уменьшая вводимое значение на 1, находим, что при 21 программа выводит 64, а при 20 – уже 128
- Ответ: 21.

6

Определите, при каком наименьшем введённом значении переменной s программа выведет число 64. Для Вашего удобства программа представлена на четырёх языках программирования.

| Паскаль | Python | Алгоритмический язык |
|---|---|--|
| <pre>var s, n: integer; begin readln (s); n := 1; while s < 51 do begin s := s + 5; n := n * 2 end; writeln(n) end.</pre> | <pre>s = int(input()) n = 1 while s < 51: s = s + 5 n = n * 2 print(n)</pre> | <pre>алг нач цел n, s ввод s n := 1 нц пока s < 51 s := s + 5 n := n * 2 кц вывод n кон</pre> |
| C++ | | |
| <pre>#include <iostream> using namespace std; int main() { int s, n; cin >> s; n = 1; while (s < 51) { s = s + 5; n = n * 2; } cout << n << endl; return 0; }</pre> | | |

Ответ: _____.

ПЕРЕБОР С ПОМОЩЬЮ ПРОГРАММЫ

- На каждой итерации цикла значение s увеличивается, и цикл заканчивается, когда s станет больше или равно 51

$s0 := 50$; // начальное значение s должно быть меньше, чем 51

while True do begin // организуем бесконечный цикл
 $s := s0$;

$n := 1$;

while $s < 51$ do begin

$s := s + 5$;

$n := n * 2$

end;

 if $n = 64$ then writeln($s0$); // выводим его на экран в том случае, если в результате работы исходного алгоритма получилось число 64

$s0 := s0 - 1$; // уменьшаем начальное значение $s0$
end;

- Программа будет последовательно выводить все числа, при вводе которых исходный алгоритм печатает 64, и последнее выведенное число и будет ответом к задаче
- Обязательно использовать новую переменную $s0$, поскольку переменная s изменяется в ходе работы внутреннего цикла

6

Определите, при каком наименьшем введённом значении переменной s программа выведет число 64. Для Вашего удобства программа представлена на четырёх языках программирования.

| Паскаль | Python | Алгоритмический язык |
|---|---|--|
| <pre>var s, n: integer; begin readln (s); n := 1; while s < 51 do begin s := s + 5; n := n * 2 end; writeln(n) end.</pre> | <pre>s = int(input()) n = 1 while s < 51: s = s + 5 n = n * 2 print(n)</pre> | <pre>алг нач цел n, s ввод s n := 1 нц пока s < 51 s := s + 5 n := n * 2 кц вывод n кон</pre> |
| C++ <pre>#include <iostream> using namespace std; int main() { int s, n; cin >> s; n = 1 ; while (s < 51) { s = s + 5; n = n * 2; } cout << n << endl; return 0; }</pre> | | |

Ответ: _____.

ПЕРЕБОР С ПОМОЩЬЮ ЭЛЕКТРОННЫХ ТАБЛИЦ

- анализ программы:
 - на каждой итерации цикла значение переменной s увеличивается на 5, а значение переменной n умножается на 2
 - цикл останавливается, когда значение s станет больше или равно 51
- вводим начальные значения (для s пока берём любое значение, например, 1):

| | A | B |
|---|---|---|
| 1 | s | n |
| 2 | 1 | 1 |

- вводим формулы, описывающие законы изменения переменных:

| | A | B |
|---|-------|-------|
| 1 | s | n |
| 2 | 1 | 1 |
| 3 | =A2+5 | =B2*2 |

- протягиваем формулы вниз, пока последнее значение в столбце n не станет равно 64:

| | A | B |
|---|----|----|
| 1 | s | n |
| 2 | 1 | 1 |
| 3 | 6 | 2 |
| 4 | 11 | 4 |
| 5 | 16 | 8 |
| 6 | 21 | 16 |
| 7 | 26 | 32 |
| 8 | 31 | 64 |

- в столбце A для переменной s полезно установить условное форматирование: при значениях, меньших, чем 51, ячейка выделяется красным цветом:

| | A | B |
|---|----|----|
| 1 | s | n |
| 2 | 1 | 1 |
| 3 | 6 | 2 |
| 4 | 11 | 4 |
| 5 | 16 | 8 |
| 6 | 21 | 16 |
| 7 | 26 | 32 |
| 8 | 31 | 64 |

- перебором находим минимальное значение ячейки A2, при котором красная область закончится сразу над зелёной ячейкой

| | A | B |
|---|----|----|
| 1 | s | n |
| 2 | 21 | 1 |
| 3 | 26 | 2 |
| 4 | 31 | 4 |
| 5 | 36 | 8 |
| 6 | 41 | 16 |
| 7 | 46 | 32 |
| 8 | 51 | 64 |

- такая ситуация будет для значений s от 21 до 25, минимальное из них – 21
- **Ответ: 21.**

ПРИМЕРЫ ФОРМУЛИРОВОК

- При каком (наименьшем) наибольшем введенном числе d после выполнения программы будет напечатано 55?
- Запишите число, которое будет напечатано в результате выполнения программы
- Запишите через запятую наименьшее и наибольшее значение числа d , которое нужно ввести, чтобы после выполнения программы было напечатано 153?
- Сколько различных значений числа d можно ввести, чтобы после выполнения программы было напечатано 246?
- Определите, при каком наименьшем (наибольшем) введенном значении переменной s программа выведет число 128.
- Сколько существует различных значений d , оканчивающихся на 8, при вводе которых эта приведенная программа выведет число 50?
- Определите, при каком наименьшем введенном значении переменной s программа выведет число, большее 100.
- Определите, при каком наименьшем положительном введенном значении переменной s программа выведет отрицательное число.
- Определите, при каком наименьшем положительном введенном значении переменной s программа выведет четырехзначное число.
- Определите, при каком наименьшем положительном введенном значении переменной s программа выведет число s без изменения его значения

113) (А.Г. Минак) Определите, при каком наибольшем положительном введённом значении переменной s программа выведет трёхзначное число.

```
var s, n: integer;  
begin  
  readln (s);  
  n := 200;  
  while s div n >= 2 do  
  begin  
    s := s + 5;  
    n := n + 5  
  end;  
  writeln(s)  
end.
```

```
#include <iostream>  
using namespace std;  
int main() {  
  int s,s0, n;  
  for (s0=800; s0>400; s0--)  
  { s=s0; n = 200;  
    while (s / n >= 2) {  
      s = s + 5;  
      n = n + 5; }  
    cout << s0 << ' ' << s << endl;  
  }  
  return 0;}
```

| s | n | s/n |
|------|-----|-----|
| 699 | 200 | 3 |
| 704 | 205 | 3 |
| 709 | 210 | 3 |
| 714 | 215 | 3 |
| 719 | 220 | 3 |
| 724 | 225 | 3 |
| 729 | 230 | 3 |
| 734 | 235 | 3 |
| 739 | 240 | 3 |
| 744 | 245 | 3 |
| 749 | 250 | 2 |
| 754 | 255 | 2 |
| 759 | 260 | 2 |
| 764 | 265 | 2 |
| 949 | 450 | 2 |
| 954 | 455 | 2 |
| 959 | 460 | 2 |
| 964 | 465 | 2 |
| 969 | 470 | 2 |
| 974 | 475 | 2 |
| 979 | 480 | 2 |
| 984 | 485 | 2 |
| 989 | 490 | 2 |
| 994 | 495 | 2 |
| 999 | 500 | 1 |
| 1004 | 505 | 1 |

ЗАДАНИЕ 17 (ПОВЫШЕННЫЙ УРОВЕНЬ, ВРЕМЯ – 15 МИН)

Тема: Перебор целых чисел на заданном отрезке.

Проверка делимости

Что проверяется:

Умение создавать собственные программы (20–40 строк) для обработки целочисленной информации.

1.7.2. Основные конструкции языка

программирования. Система программирования.

1.1.5. Умение создавать программы на языке программирования по их описанию.

17

Рассматривается множество целых чисел, принадлежащих числовому отрезку $[1016; 7937]$, которые делятся на 3 и не делятся на 7, 17, 19, 27.

Найдите количество таких чисел и максимальное из них.

В ответе запишите два целых числа: сначала количество, затем максимальное число.

Для выполнения этого задания можно написать программу или воспользоваться редактором электронных таблиц.

Ответ:

| | |
|----------------------|----------------------|
| <input type="text"/> | <input type="text"/> |
|----------------------|----------------------|

ЗАДАНИЕ 17 (ПОВЫШЕННЫЙ УРОВЕНЬ, ВРЕМЯ – 15 МИН)

Что нужно знать:

- время выполнения несущественно для отрезков, заданных для перебора, поэтому можно использовать простой перебор без оптимизации
- общая структура цикла перебора всех целых чисел на отрезке $[a; b]$ и подсчета количества, для которых выполняется некоторое условие

Python:

```
count = 0
for n in range(a, b+1):
    if условие выполнено:
        count += 1
print( count )
```

Pascal:

```
count := 0;
for n:=a to b do
    if условие выполнено then
        count := count + 1;
writeln(count);
```

C++:

```
int count = 0;
for(int n = a; n <= b; n++)
    if( условие выполнено )
        count += 1;
std::cout << count;
```

ЗАДАНИЕ 17 (ПОВЫШЕННЫЙ УРОВЕНЬ, ВРЕМЯ – 15 МИН)

- проверить делимость числа n на число d можно с помощью операции взятия остатка от деления n на d : если остаток равен 0, число n делится на d нацело

Python:

```
if n % d == 0:  
    print("Делится")  
else: print("Не делится")
```

Pascal:

```
if n mod d = 0 then  
    writeln('Делится')  
else writeln('Не делится')
```

C++:

```
if( n % d == 0 )  
    std::cout << "Делится";  
else std::cout << "Не делится";
```

СПОСОБЫ РЕШЕНИЯ

- с помощью электронных таблиц
- с собственной программы

17

Рассматривается множество целых чисел, принадлежащих числовому отрезку $[1016; 7937]$, которые делятся на 3 и не делятся на 7, 17, 19, 27.

Найдите количество таких чисел и максимальное из них.

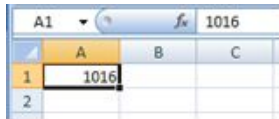
В ответе запишите два целых числа: сначала количество, затем максимальное число.

Для выполнения этого задания можно написать программу или воспользоваться редактором электронных таблиц.

Ответ:

С ПОМОЩЬЮ ЭЛЕКТРОННЫХ ТАБЛИЦ

- введём начальное число в ячейку A1:

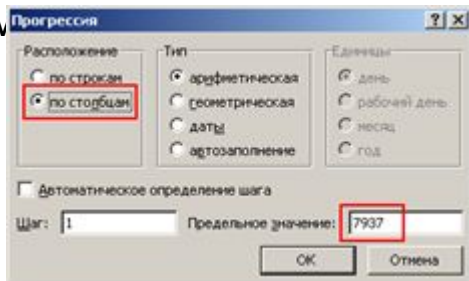


| | |
|----|------|
| A1 | 1016 |
|----|------|

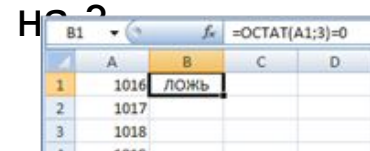
- заполним ряд натуральных чисел до конечного числа; на вкладке Главная выберем команду Прогрессия:



- введём шаг и конечное значение, заполнение по столбцам



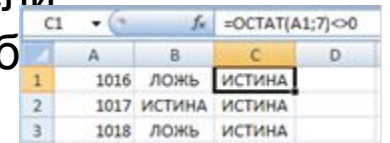
- в столбце определим логическое значение: истина, если остаток от деления числа в столбце A делится



| | | |
|----|----------------|------|
| B1 | =ОСТАТ(A1;3)=0 | |
| A | 1016 | ЛОЖЬ |
| A | 1017 | |
| A | 1018 | |

- двойным щелчком по маркеру заполнения скопируем формулу на весь столбец (пока не кончатся данные в столбце A)

- в столбце C выведем ИСТИНА, если соответствующее значение в столбце B равно 1:



| | | | |
|----|----------------|--------|--------|
| C1 | =ОСТАТ(A1;7)=0 | | |
| A | 1016 | ЛОЖЬ | ИСТИНА |
| A | 1017 | ИСТИНА | ИСТИНА |
| A | 1018 | ЛОЖЬ | ИСТИНА |

(Б.С. Михлин) Можно записать формулу чуть короче: =ОСТАТ(A1;7), т.к. численное значение, отличное от нуля, рассматривается как ИСТИНА

С ПОМОЩЬЮ ЭЛЕКТРОННЫХ ТАБЛИЦ

- в столбцах D, E, F аналогично выведем ИСТИНА, если соответствующее значение в столбце A не делится на 17, 19 и 27:

| | A | B | C | D | E | F |
|---|------|--------|--------|--------|--------|--------|
| 1 | 1016 | ЛОЖЬ | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА |
| 2 | 1017 | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА |
| 3 | 1018 | ЛОЖЬ | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА |
| 4 | 1019 | ЛОЖЬ | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА |

- в столбце G используем функцию ЕСЛИ; если все значения ячеек в столбцах B-F в этой строке истинны, выводим число из A1, иначе – пустую строку:

| | A | B | C | D | E | F | G |
|---|------|--------|--------|--------|--------|--------|------|
| 1 | 1016 | ЛОЖЬ | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА | |
| 2 | 1017 | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА | 1017 |
| 3 | 1018 | ЛОЖЬ | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА | |
| 4 | 1019 | ЛОЖЬ | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА | |

(Б.С. Михлин) Можно написать формулу чуть короче, используя диапазон в команде И: =ЕСЛИ(И(B1:F1);A1;"")

- теперь в столбце G мы видим только те числа, которые удовлетворяют условию задачи; прокрутив таблицу вниз, узнаем, что последняя строка имеет номер 6922, поэтому находим количество и максимум для диапазона G1:G6922 в любых свободных ячейках:

=СЧЁТ(G1:G6922)

=МАКС(G1:G6922) (это последнее число в столбце G)

| | A | B | C | D | E | F | G | H |
|---|------|--------|--------|--------|--------|--------|------|------|
| 1 | 1016 | ЛОЖЬ | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА | | 1568 |
| 2 | 1017 | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА | 1017 | 7935 |
| 3 | 1018 | ЛОЖЬ | ИСТИНА | ИСТИНА | ИСТИНА | ИСТИНА | | |

- Ответ: 1568 7935

(И.В. Степанов) чтобы уменьшить таблицу в 3 раза, можно проверять только числа, кратные 3, с шагом 3; первое число на заданном отрезке, кратное 3 – это 1017 (сумма цифр делится на 3).

РЕШЕНИЕ С ПОМОЩЬЮ ПРОГРАММЫ

```
#include <iostream>
int main()
{
    int count = 0;
    int maxGood = 0;
    for (int n=1016; n<=7937; n++)
        if( (n % 3 == 0) and (n % 7 != 0) and
            (n % 17 != 0) and (n % 19 != 0) and (n %
27 != 0) ) {
            maxGood = n;
            count += 1;
        }
    std::cout << count << " " << maxGood;
}
```

```
var count, n, maxGood: integer;
begin
    count:= 0;
    maxGood:= 0;
    for n:=1016 to 7937 do
        if (n mod 3 = 0) and (n mod 7 <> 0) and
            (n mod 17 <> 0) and (n mod 19 <> 0)
and
            (n mod 27 <> 0) then begin
                maxGood:= n;
                count := count + 1
            end;
        writeln(count, ' ', maxGood)
    end.
```

28) (А. Куканова) Рассматривается множество целых чисел, принадлежащих числовому отрезку [3439; 7410], которые удовлетворяют следующим условиям:

- запись в двоичной и шестеричной системах счисления заканчивается разными цифрами;
- кратны, по крайней мере, одному из чисел: 9, 10 или 11.

Найдите количество таких чисел и максимальное из них.

| | A | B | C | D | E | F | G | H | I | J |
|---|------|--------------|--------------|---------|----------------|-----------------|-----------------|-------------------------------|-----------------|-----------------|
| 1 | 3439 | =ОСТАТ(A1;2) | =ОСТАТ(A1;6) | =B1<>C1 | =ОСТАТ(A1;9)=0 | =ОСТАТ(A1;10)=0 | =ОСТАТ(A1;11)=0 | =ЕСЛИ(И(D1;ИЛИ(E1:G1));A1;"") | =СЧЁТ(H1:H3972) | =МАКС(H1:H3972) |
| 2 | 3440 | =ОСТАТ(A2;2) | =ОСТАТ(A2;6) | =B2<>C2 | =ОСТАТ(A2;9)=0 | =ОСТАТ(A2;10)=0 | =ОСТАТ(A2;11)=0 | =ЕСЛИ(И(D2;ИЛИ(E2:G2));A2;"") | | |
| 3 | 3441 | =ОСТАТ(A3;2) | =ОСТАТ(A3;6) | =B3<>C3 | =ОСТАТ(A3;9)=0 | =ОСТАТ(A3;10)=0 | =ОСТАТ(A3;11)=0 | =ЕСЛИ(И(D3;ИЛИ(E3:G3));A3;"") | | |
| 4 | 3442 | =ОСТАТ(A4;2) | =ОСТАТ(A4;6) | =B4<>C4 | =ОСТАТ(A4;9)=0 | =ОСТАТ(A4;10)=0 | =ОСТАТ(A4;11)=0 | =ЕСЛИ(И(D4;ИЛИ(E4:G4));A4;"") | | |
| 5 | 3443 | =ОСТАТ(A5;2) | =ОСТАТ(A5;6) | =B5<>C5 | =ОСТАТ(A5;9)=0 | =ОСТАТ(A5;10)=0 | =ОСТАТ(A5;11)=0 | =ЕСЛИ(И(D5;ИЛИ(E5:G5));A5;"") | | |

```
#include <iostream>
int main(){
    int count = 0;
    int maxGood = 0;
    for (int n=3439; n<=7410; n++)
        if((n % 2 != n % 6) and
            ((n % 9 == 0) or (n % 10 == 0) or (n % 11 == 0) )) {
            maxGood = n;      count += 1;    }
    std::cout << count << " " << maxGood;}
```


ВОЗМОЖНЫЕ ВАРИАНТЫ ФОРМУЛИРОВКИ УСЛОВИЙ

- запись в **двоичной** и **шестеричной** системах счисления **заканчивается разными (одинаковыми) цифрами**
- запись в **двоичной** системе заканчивается на **11**
- запись которых в **пятеричной** системе **имеет не менее 6 цифр** и **заканчивается на 21 или 23**
- **не** делятся нацело на 5, 7 и 11 И запись в троичной системе счисления **имеет ровно 8 цифр**
- кратны 2, но не кратны 16 И **цифра в разряде десятков не менее 4**
- цифра в разряде десятков принадлежит отрезку $[3; 7]$ И цифра в разряде сотен отлична от 1 и 9.
- количество цифр в восьмеричной и десятичной записях числа не совпадает И остаток от деления на 7 равен 1 или 5.

ЗАДАНИЕ 24 (ВЫСОКИЙ УРОВЕНЬ, ВРЕМЯ – 18 МИНУТ)

Тема: Обработка символьных строк

Что проверяется:

Умение создавать собственные программы (10–20 строк) для обработки символьной информации.

1.5.2. Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы), псевдослучайные последовательности.

1.1.3. Строить информационные модели объектов, систем и процессов в виде алгоритмов.



Задание выполняется с использованием прилагаемых файлов.

24

Текстовый файл состоит не более чем из 10^6 символов X, Y и Z. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны. Для выполнения этого задания следует написать программу.

Ответ: _____.

ЧТЕНИЕ СТРОК ИЗ ФАЙЛА

в языке C++ используются потоки:

```
#include <fstream>
#include <string>
using namespace std;
int main()
{
    ifstream F("k7.txt");
    string s;
    getline(F, s);
    ...
}
```

в языке Python удобнее всего использовать такую конструкцию:

```
with open("k7.txt", "r") as F:
    s = F.readline()
```

конструкция **with-as** – это *контекстный менеджер*, в данном случае он открывает указанный файл в режиме чтения (второй аргумент «r» при вызове функции **open**), записывает ссылку на него в файловую переменную **F**, выполняет тело блока (читает первую строку файла в переменную **s**) и закрывает (освобождает) файл

равносилен такому

```
F = open("k7.txt")
... # какие-то операции с файлом
F.close()
```

ЧТЕНИЕ СТРОК ИЗ ФАЙЛА

В языке PascalABC.NET можно выполнить перенаправление потока ввода:

```
assign( input, 'k7.txt' );  
readln(s);
```

программа будет «думать», что читает данные, введённые с клавиатуры (с консоли), а на самом деле эти данные будут прочитаны из файла **k7.txt**

В языке FreePascal также можно выполнить перенаправление потока ввода, но нужно дополнительно открывать входной поток:

```
assign( input, 'k7.txt' );  
reset( input );      { для FreePascal!!! }  
readln(s);
```

при работе в среде FreePascal нужно убедиться, что в параметрах компилятора включена поддержка **длинных символьных строк**; на всякий случай стоит добавить в первой строке программы директиву
{\$H+}

Среда PascalABC НЕ ПОДДЕРЖИВАЕТ работу с **длинными символьными строками**, поэтому для решения задачи использовать версию PascalABC.NET, которую можно бесплатно скачать с сайта автора www.pascalabc.net.

ПОИСК САМОЙ ДЛИННОЙ ЦЕПОЧКИ СИМВОЛОВ «С»

cLen – длина текущей цепочки букв С

maxLen – максимальная длина цепочки букв С на данный момент

рассмотрим очередной символ строки; если это буква С, увеличиваем **cLen** на 1 и, если нужно запоминаем новую максимальную длину; если это не буква С, просто записываем с **cLen** ноль:

maxLen = 0

cLen = 0

for c in s:

обработать текущий символ

if c == 'C':

cLen += 1 ещё одна буква С

if cLen > maxLen: **# возможно, новая максимальная длина**

maxLen = cLen

else:

cLen = 0 **# цепочка букв С кончилась**

ПОИСК САМОЙ ДЛИННОЙ ЦЕПОЧКИ ЛЮБЫХ СИМВОЛОВ

curLen – длина текущей цепочки одинаковых символов

maxLen – максимальная длина цепочки одинаковых символов на данный момент

c – символ, из которого строится самая длинная подцепочка

в начальный момент рассмотрим один первый символ (цепочка длины 1 есть всегда!):

```
maxLen = 1
```

```
curLen = 1
```

```
c = s[0]
```

```
for i in range(1, len(s)): # в цикле, начиная со второго по счёту до конца строки, обработать пару символов s[i-1] и s[i]
```

```
if s[i] == s[i-1]: # если очередной символ s[i] такой же, как и предыдущий, то цепочка продолжается
```

```
    curLen += 1 # увеличиваем длину
```

```
    if curLen > maxLen: # если цепочка побила рекорд
```

```
        maxLen = curLen # запоминаем её длину...
```

```
        c = s[i] # и новый базовый символ в переменной c
```

```
else:
```

```
    curLen = 1 # началась новая цепочка
```

если очередной символ не совпал с предыдущим, началась новая цепочка, и её длина пока равна 1 (это значение записывается в переменную curLen)

P-01. В ТЕКСТОВОМ ФАЙЛЕ k7.txt НАХОДИТСЯ ЦЕПОЧКА ИЗ СИМВОЛОВ ЛАТИНСКОГО АЛФАВИТА A, B, C. НАЙДИТЕ ДЛИНУ САМОЙ ДЛИННОЙ ПОДЦЕПОЧКИ, СОСТОЯЩЕЙ ИЗ СИМВОЛОВ C.

решение на Паскале:

```
var maxLen, cLen, i: integer;
    s: string;
begin
    assign(input, 'k7.txt');
    readln(s);
    maxLen := 0;
    cLen := 0;
    for i:=1 to Length(s) do
        if s[i] = 'C' then begin
            cLen := cLen + 1;
            if cLen > maxLen then maxLen := cLen;
        end
    else
        cLen := 0;
    writeln(maxLen);
end.
end.
```

решение на Python:

```
with open( "k7.txt", "r" ) as F:
    s = F.readline()
    maxLen, cLen = 0, 0
    for c in s:
        if c == 'C':
            cLen += 1
            if cLen > maxLen:
                maxLen = cLen
        else:
            cLen = 0
    print( maxLen )
```

P-01. В ТЕКСТОВОМ ФАЙЛЕ k7.txt НАХОДИТСЯ ЦЕПОЧКА ИЗ СИМВОЛОВ ЛАТИНСКОГО АЛФАВИТА A, B, C. НАЙДИТЕ ДЛИНУ САМОЙ ДЛИННОЙ ПОДЦЕПОЧКИ, СОСТОЯЩЕЙ ИЗ СИМВОЛОВ C.

Встроенная функция поиска подстроки: ищем строку из одного символа C, потом – из двух символов, из трёх и т. д.; в какой-то момент поиск не даст результата, значит ответ – это длина предыдущей цепочки, которая короче текущей на единицу

решение на Паскале:

```
var cc, s: string;

begin

  assign(input, 'k7.txt');

  readln(s);

  cc := 'C';

  while Pos(cc, s) > 0 do

    cc := cc + 'C';

  writeln( Length(cc)-1 );

end.
```

решение на Python:

```
with open( "k7.txt", "r" ) as F:

  s = F.readline()

  cc = 'C'

  while cc in s:

    cc += 'C'

  print( len(cc)-1 )
```


R-01. В ТЕКСТОВОМ ФАЙЛЕ k7.txt НАХОДИТСЯ ЦЕПОЧКА ИЗ СИМВОЛОВ ЛАТИНСКОГО АЛФАВИТА A, B, C. НАЙДИТЕ ДЛИНУ САМОЙ ДЛИННОЙ ПОДЦЕПОЧКИ, СОСТОЯЩЕЙ ИЗ СИМВОЛОВ C.

1. Команда Найти
2. Встроенные текстовые функции электронных таблиц: FIND (НАЙТИ) или SEARCH (ПОИСК) и REPT (ПОВТОР)

Функции НАЙТИ и ПОИСК выводят положение начала искомой подцепочки в заданной цепочке символов или сообщение #ЗНАЧ!, если подцепочка не найдена. Если поиск надо осуществлять с начала цепочки, то третий параметр функций НАЙТИ и ПОИСК можно не указывать. Функция НАЙТИ учитывает регистр символов. Функция ПОИСК не учитывает регистр символов и в ней можно использовать подстановочные символы (* и ?).

| | A | B |
|---|-----------------------------|---|
| 1 | | |
| 2 | BACBAAAAAACCCAAAABBABABABV | |
| 3 | =НАЙТИ(ПОВТОР("C";B3);A\$2) | 1 |

| | | |
|---|----------------------------|----|
| | | |
| | BACBAAAAAACCCAAAABBABABABV | |
| | 3 | 1 |
| | 11 | 2 |
| ✓ | #ЗНАЧ! | 3 |
| ✓ | #ЗНАЧ! | 4 |
| ✓ | #ЗНАЧ! | 5 |
| ✓ | #ЗНАЧ! | 6 |
| ✓ | #ЗНАЧ! | 7 |
| ✓ | #ЗНАЧ! | 8 |
| ✓ | #ЗНАЧ! | 9 |
| ✓ | #ЗНАЧ! | 10 |
| ✓ | #ЗНАЧ! | 11 |
| ✓ | #ЗНАЧ! | 12 |
| ✓ | #ЗНАЧ! | 13 |
| ✓ | #ЗНАЧ! | 14 |
| ✓ | #ЗНАЧ! | 15 |
| ✓ | #ЗНАЧ! | 16 |
| ✓ | #ЗНАЧ! | 17 |
| ✓ | #ЗНАЧ! | 18 |
| ✓ | #ЗНАЧ! | 19 |
| ✓ | #ЗНАЧ! | 20 |
| ✓ | #ЗНАЧ! | 21 |
| ✓ | #ЗНАЧ! | 22 |

P-06. В ТЕКСТОВОМ ФАЙЛЕ **k8.txt** НАХОДИТСЯ ЦЕПОЧКА ИЗ СИМВОЛОВ, В КОТОРУЮ МОГУТ ВХОДИТЬ ЗАГЛАВНЫЕ БУКВЫ ЛАТИНСКОГО АЛФАВИТА А...Z И ДЕСЯТИЧНЫЕ ЦИФРЫ. НАЙДИТЕ ДЛИНУ САМОЙ ДЛИННОЙ ПОДЦЕПОЧКИ, СОСТОЯЩЕЙ ИЗ ОДИНАКОВЫХ СИМВОЛОВ. **ДЛЯ КАЖДОЙ ЦЕПОЧКИ МАКСИМАЛЬНОЙ ДЛИНЫ** ВЫВЕДИТЕ В ОТДЕЛЬНОЙ СТРОКЕ СНАЧАЛА СИМВОЛ, ИЗ КОТОРОГО СТРОИТСЯ ЭТА ЦЕПОЧКА, А ЗАТЕМ ЧЕРЕЗ ПРОБЕЛ – ДЛИНУ ЭТОЙ ЦЕПОЧКИ.

```
// ЯЗЫК PascalABC.NET
```

```
var maxLen, curLen, i: integer;
```

```
    s: string; // не имеет ограничения на длину строк; в устаревших версиях языка Pascal длина строки не может превышать 255 символов
```

```
begin
```

```
    assign(input, 'k8.txt'); // перенаправление входного потока с консоли на файл
```

```
    readln(s);
```

```
    maxLen := 1;
```

```
    curLen := 1;
```

```
    var c := new List<char>; // создаем динамический массив (список), тип данных List (список)
```

```
    c.Add( s[1] ); //записываем первый символ
```

```
    for i:=2 to Length(s) do // в цикле со второго символа до конца строки
```

```
        if s[i] = s[i-1] then begin // если текущий и предыдущий символы одинаковы
```

```
            curLen := curLen + 1; // увеличиваем длину подстроки
```

```
            if curLen = maxLen then begin // если нашли новую (не первую) цепочку максимальной длины, добавляем новый символ в список:
```

```
                c.Add( s[i] );
```

```
            end
```

```
            else if curLen > maxLen then begin // если нашли новую самую длинную цепочку с длиной большей, чем все предыдущие, очищаем список и добавляем в него нов
```

```
СИМВОЛ.
```

```
                maxLen := curLen;
```

```
                c.Clear;
```

```
                c.Add( s[i] );
```

```
            end
```

```
        end
```

```
    else
```

```
        curLen := 1;
```

```
    foreach var c1 in c do // после окончания обработки нужно вывести все символы и длины цепочек, удобнее всего использовать для этого цикл foreach
```

```
        writeln( c1, ' ', maxLen );
```

```
end.
```

P-06. В ТЕКСТОВОМ ФАЙЛЕ **k8.txt** НАХОДИТСЯ ЦЕПОЧКА ИЗ СИМВОЛОВ, В КОТОРУЮ МОГУТ ВХОДИТЬ ЗАГЛАВНЫЕ БУКВЫ ЛАТИНСКОГО АЛФАВИТА А...Z И ДЕСЯТИЧНЫЕ ЦИФРЫ. НАЙДИТЕ ДЛИНУ САМОЙ ДЛИННОЙ ПОДЦЕПОЧКИ, СОСТОЯЩЕЙ ИЗ ОДИНАКОВЫХ СИМВОЛОВ. **ДЛЯ КАЖДОЙ ЦЕПОЧКИ МАКСИМАЛЬНОЙ ДЛИНЫ** ВЫВЕДИТЕ В ОТДЕЛЬНОЙ СТРОКЕ СНАЧАЛА СИМВОЛ, ИЗ КОТОРОГО СТРОИТСЯ ЭТА ЦЕПОЧКА, А ЗАТЕМ ЧЕРЕЗ ПРОБЕЛ – ДЛИНУ ЭТОЙ ЦЕПОЧКИ.

программа на языке Python:

```
s = open('k7.txt').read()
count = 0
maxCount = 0
for char in s:
    if (char == 'E' and count%3 == 0) or \
        (char == 'A' and count%3 == 1) or \
        (char == 'B' and count%3 == 2):
        count += 1
        if count > maxCount:
            maxCount = count
    elif (char=='E'):
        count = 1
    else:
        count = 0
print(maxCount)
```

ЗАДАНИЕ 25 (ВЫСОКИЙ УРОВЕНЬ, ВРЕМЯ – 20 МИНУТ)

Тема: Обработка целых чисел. Проверка делимости

Что проверяется:

Умение создавать собственные программы (10–20 строк) для обработки целочисленной информации.

1.6.3. Построение алгоритмов и практические вычисления

1.1.5. Создавать программы на языке программирования по их описанию

25

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку $[174457; 174505]$, числа, имеющие ровно два различных натуральных делителя, не считая единицы и самого числа. Для каждого найденного числа запишите эти два делителя в таблицу на экране с новой строки в порядке возрастания произведения этих двух делителей. Делители в строке таблицы также должны следовать в порядке возрастания.

Например, в диапазоне $[5; 9]$ ровно два целых различных натуральных делителя имеют числа 6 и 8, поэтому для этого диапазона таблица на экране должна содержать следующие значения:

| | |
|---|---|
| 2 | 3 |
| 2 | 4 |

Ответ:

| | |
|-----|-----|
| | |
| ... | ... |
| | |

РЕШЕНИЕ С ПОМОЩЬЮ ПРОГРАММЫ

- время выполнения несущественно для отрезков, поэтому можно использовать простой перебор без оптимизации
- общая структура цикла перебора

Pascal:

```
count := 0;
for n:=a to b do
  if условие выполнено then
    действие // например, подсчет количества count := count +
1;
writeln(count);
```

C++:

```
int count = 0;
for(int n = a; n <= b; n++)
  if( условие выполнено )
    действие;
std::cout << count;
```

Python:

```
count = 0
for n in range(a, b+1):
  if условие выполнено:
    действие
print( count )
```

25

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [174457; 174505], числа, имеющие ровно два различных натуральных делителя, не считая единицы и самого числа. Для каждого найденного числа запишите эти два делителя в таблицу на экране с новой строки в порядке возрастания произведения этих двух делителей. Делители в строке таблицы также должны следовать в порядке возрастания.

Например, в диапазоне [5; 9] ровно два целых различных натуральных делителя имеют числа 6 и 8, поэтому для этого диапазона таблица на экране должна содержать следующие значения:

| | |
|---|---|
| 2 | 3 |
| 2 | 4 |

Ответ:

| | |
|-----|-----|
| ... | ... |
| ... | ... |

РЕШЕНИЕ С ПОМОЩЬЮ ПРОГРАММЫ

- Количество делителей
- Количество делителей и сами делители
- Простые числа - любое простое число имеет только два делителя
- Перебор всех возможных делителей d от 1 до n
- Сохранение в массиве найденных делителей
- Оптимизация перебора: наименьший из пары делителей, таких что $a \cdot b = n$, не превышает квадратного корня из n ; нужно только аккуратно обработать случай, когда число n представляет собой квадрат другого целого числа

P-02 (ДЕМО-2021). НАПИШИТЕ ПРОГРАММУ, КОТОРАЯ ИЩЕТ СРЕДИ ЦЕЛЫХ ЧИСЕЛ, ПРИНАДЛЕЖАЩИХ ЧИСЛОВОМУ ОТРЕЗКУ [174457; 174505], ЧИСЛА, ИМЕЮЩИЕ РОВНО ДВА РАЗЛИЧНЫХ НАТУРАЛЬНЫХ ДЕЛИТЕЛЯ, НЕ СЧИТАЯ ЕДИНИЦЫ И САМОГО ЧИСЛА. ДЛЯ КАЖДОГО НАЙДЕННОГО ЧИСЛА ЗАПИШИТЕ ЭТИ ДВА ДЕЛИТЕЛЯ В ТАБЛИЦУ НА ЭКРАНЕ С НОВОЙ СТРОКИ В ПОРЯДКЕ ВОЗРАСТАНИЯ ПРОИЗВЕДЕНИЯ ЭТИХ ДВУХ ДЕЛИТЕЛЕЙ. ДЕЛИТЕЛИ В СТРОКЕ ТАБЛИЦЫ ТАКЖЕ ДОЛЖНЫ СЛЕДОВАТЬ В ПОРЯДКЕ ВОЗРАСТАНИЯ.

программа на **Паскале**:

```
const divCount = 2;
var n, count, d, i, j, q: longint;
    divs: array[1..divCount] of longint;
begin
  for n:=174457 to 174505 do begin
    count := 0;
    q := floor(sqrt(n));
    for d:=2 to q do
      if n mod d = 0 then begin
        count := count + 2;
        if count <= divCount then begin
          divs[count-1] := d;
          if d <> n div d then
            divs[count] := n div d;
          end
        else break
      end;
    if count = divCount then begin
      for i:=1 to divCount do
        write(divs[i], ' ');
      writeln
    end
  end
end.
```

программа на **C++**:

```
#include <iostream>
#include <cmath>
int main()
{
  const int divCount = 2;
  int divs[divCount] = {};
  for( int n = 174457; n <= 174505; n++ ) {
    int count = 0;
    int q = int(sqrt(n));
    for( int d = 2; d <= q; d++ )
      if( n % d == 0 ) {
        count += 2;
        if( count <= divCount ) {
          divs[count-2] = d;
          if( d != n / d )
            divs[count-1] = n / d;
        }
        else break;
      }
    if( count == divCount ) {
      for( int i = 0; i < divCount; i++ )
        std::cout << divs[i] << ' ';
      std::cout << std::endl;
    }
  }
}
```

P-02 (ДЕМО-2021). НАПИШИТЕ ПРОГРАММУ, КОТОРАЯ ИЩЕТ СРЕДИ ЦЕЛЫХ ЧИСЕЛ, ПРИНАДЛЕЖАЩИХ ЧИСЛОВОМУ ОТРЕЗКУ [174457; 174505], ЧИСЛА, ИМЕЮЩИЕ РОВНО ДВА РАЗЛИЧНЫХ НАТУРАЛЬНЫХ ДЕЛИТЕЛЯ, НЕ СЧИТАЯ ЕДИНИЦЫ И САМОГО ЧИСЛА. ДЛЯ КАЖДОГО НАЙДЕННОГО ЧИСЛА ЗАПИШИТЕ ЭТИ ДВА ДЕЛИТЕЛЯ В ТАБЛИЦУ НА ЭКРАНЕ С НОВОЙ СТРОКИ В ПОРЯДКЕ ВОЗРАСТАНИЯ ПРОИЗВЕДЕНИЯ ЭТИХ ДВУХ ДЕЛИТЕЛЕЙ. ДЕЛИТЕЛИ В СТРОКЕ ТАБЛИЦЫ ТАКЖЕ ДОЛЖНЫ СЛЕДОВАТЬ В ПОРЯДКЕ ВОЗРАСТАНИЯ.

программа на **Паскале**:

```
var i, j, k, d1, d2: longint;
begin
  for i:=174457 to 174505 do
  begin
    k := 0;
    for j:= 2 to i-1 do
      if i mod j = 0 then begin
        k:= k + 1;
        if k = 1 then d1:= j;
          if k = 2 then d2:= j;
        end;
      if k = 2 then
        writeln( d1, ' ', d2 );
      end;
    end.
  end.
```

программа на **C++**:

```
#include <iostream>
using namespace std;
int main()
{
  int d1, d2;
  for( int i = 174457; i <= 174505;
  i++ ) {
    int k = 0;
    for( int j = 2; j <= i-1; j++ )
      if( i % j == 0 ) {
        k ++;
        if( k == 1 ) d1 = j;
        if( k== 2) d2 = j;
      }
    if( k == 2 ) cout << d1 << ' ' << d2
    << endl;
  }
  return 0;
}
```

программа на **Python**:

```
for i in range (174457,
174505+1):
  k = 0;
  for j in range (2, i):
    if i % j == 0:
      k = k + 1;
      if k == 1: d1 = j
      if k == 2: d2 = j
  if k == 2:
    print( d1, d2 )
```


РЕШЕНИЕ С ПОМОЩЬЮ ЭЛЕКТРОННОЙ ТАБЛИЦЫ

1. Заполняем в строке 2 значениями из отрезка [174457;174505]
2. В столбце A заполняем делителями – это значения от 2 до $\text{sqrt}(174505)=417$, так как по нам не нужно рассматривать единицу и само число в качестве делителя
3. В ячейки по центру записываем формулу, вычисляющую остаток от деления числа из строки 2 на число из столбца A
4. В строке 419 находим количество чисел (это числа от 2 до 417), на которые число из строки 2 делится без остатка
5. Нас интересует только два различных делителя, поэтому в строке 420 для чисел, у которых только делитель ≤ 417 , находим позицию этого делителя это и есть первый делитель. Для этого используем функцию ПОИСКПОЗ; она находит в столбце 0 и определяет его позицию (третий аргумент функции ПОИСКПОЗ означает точное совпадение):
6. В строке 421 находим второй делитель числа

25 Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [174457; 174505], числа, имеющие ровно два различных натуральных делителя, не считая единицы и самого числа. Для каждого

ой пи

B3 : X ✓ fx =ОСТАТ(B\$2;\$A3)

| | A | B | C | D | E | F | G | H | I | J | ых на |
|---|--------|--------|---------|--------|--------|--------|--------|--------|--------|--------|-------|
| 1 | 174457 | 174505 | 417,738 | | | | | | | | |
| 2 | | 174457 | 174458 | 174459 | 174460 | 174461 | 174462 | 174463 | 174464 | 174465 | |
| 3 | 2 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | |
| 4 | 3 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | |
| 5 | 4 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | |
| 6 | 5 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 | 0 | |

B419 : X ✓ fx =СЧЁТЕСЛИ(B3:B418;0)

| | A | B | C | D | E | F | G | H | I | J |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 415 | 414 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 |
| 416 | 415 | 157 | 158 | 159 | 160 | 161 | 162 | 163 | 164 | 165 |
| 417 | 416 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 | 161 |
| 418 | 417 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| 419 | | 0 | 3 | 1 | 23 | 1 | 3 | 1 | 15 | 5 |
| 420 | | | | 3 | | 7 | | 59 | | |
| | | | | | | | | 57 | | |

B423 : X ✓ fx

| | A | B | C |
|-----|-----|--|--|
| 415 | 414 | =ОСТАТ(B\$2;\$A415) | =ОСТАТ(C\$2;\$A415) |
| 416 | 415 | =ОСТАТ(B\$2;\$A416) | =ОСТАТ(C\$2;\$A416) |
| 417 | 416 | =ОСТАТ(B\$2;\$A417) | =ОСТАТ(C\$2;\$A417) |
| 418 | 417 | =ОСТАТ(B\$2;\$A418) | =ОСТАТ(C\$2;\$A418) |
| 419 | | =СЧЁТЕСЛИ(B3:B418;0) | =СЧЁТЕСЛИ(C3:C418;0) |
| 420 | | =ЕСЛИ(B419=1;ПОИСКПОЗ(0;B2:B418;0);"") | =ЕСЛИ(C419=1;ПОИСКПОЗ(0;C2:C418;0);"") |
| 421 | | =ЕСЛИ(B419=1;B2/B420;"") | =ЕСЛИ(C419=1;C2/C420;"") |
| 422 | | | |

ЗАДАНИЕ 26 (ВЫСОКИЙ УРОВЕНЬ) ВРЕМЯ – 35 МИНУТ

Тема: Обработка массива целых чисел из файла. Сортировка.

Что проверяется:

Умение обрабатывать целочисленную информацию с использованием сортировки.

1.6.3. Построение алгоритмов и практические вычисления.

1.1.3. Строить информационные модели объектов, систем и процессов в виде алгоритмов.

26

Системный администратор раз в неделю создаёт архив пользовательских файлов. Однако объём диска, куда он помещает архив, может быть меньше, чем суммарный объём архивируемых файлов.

Известно, какой объём занимает файл каждого пользователя.

По заданной информации об объёме файлов пользователей и свободном объёме на архивном диске определите максимальное число пользователей, чьи файлы можно сохранить в архиве, а также максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

Входные данные.

В первой строке входного файла находятся два числа: S – размер свободного места на диске (натуральное число, не превышающее 10 000) и N – количество пользователей (натуральное число, не превышающее 1000). В следующих N строках находятся значения объёмов файлов каждого пользователя (все числа натуральные, не превышающие 100), каждое в отдельной строке.

Запишите в ответе два числа: сначала наибольшее число пользователей, чьи файлы могут быть помещены в архив, затем максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

Пример входного файла:

```
100 4
80
30
50
40
```

При таких исходных данных можно сохранить файлы максимум двух пользователей. Возможные объёмы этих двух файлов 30 и 40, 30 и 50 или 40 и 50. Наибольший объём файла из перечисленных пар – 50, поэтому ответ для приведённого примера:

| | |
|---|----|
| 2 | 50 |
|---|----|

Ответ:

| | |
|--|--|
| | |
|--|--|

РЕШЕНИЕ С ПОМОЩЬЮ ЭЛЕКТРОННОЙ ТАБЛИЦЫ

1. Перенос данных в таблицу Excel
2. Подготовка данных для обработки: сохранение нужного числа (8200), удаление лишних строк и т.п.
3. Сортировка данных
4. Выделение ячеек, сумма которых не больше 8200
5. Запоминаем количество ячеек, сумма значений которых не больше $8200 = 568$
6. Вычисление расхождения $8200 -$ текущая сумма $= 24$
7. Заменяем последнее значение большим - $29+24=53$
8. 53 нет, заменяем на **50**

26

Системный администратор раз в неделю создаёт архив пользовательских файлов. Однако объём диска, куда он помещает архив, может быть меньше, чем суммарный объём архивируемых файлов.

Известно, какой объём занимает файл каждого пользователя.

По заданной информации об объёме файлов пользователей и свободном объёме на архивном диске определите максимальное число пользователей, чьи файлы можно сохранить в архиве, а также максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

Входные данные.

В первой строке входного файла находятся два числа: S – размер свободного места на диске (натуральное число, не превышающее 10 000) и N – количество пользователей (натуральное число, не превышающее 1000). В следующих N строках находятся значения объёмов файлов каждого пользователя (все числа натуральные, не превышающие 100), каждое в отдельной строке.

Запишите в ответе два числа: сначала наибольшее число пользователей, чьи файлы могут быть помещены в архив, затем максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

Пример входного файла:

```
100 4
80
30
50
40
```

При таких исходных данных можно сохранить файлы максимум двух пользователей. Возможные объёмы этих двух файлов 30 и 40, 30 и 50 или 40 и 50. Наибольший объём файла из перечисленных пар – 50, поэтому ответ для приведённого примера:

2 50

Ответ:

РЕШЕНИЕ С ПОМОЩЬЮ ПРОГРАММЫ

Чтение данных из файла

В языке Python для чтения данных удобно использовать менеджер контекста, который открывает файл и закрывает его;

```
with open("26.txt") as Fin:
```

```
... # какие-то операции с файлом
```

если в текущей строке файла находится целое число, то прочесть его в переменную `x` можно так:

```
x = int( Fin.readline() )
```

если в строке записаны два числа, после чтения (`Fin.readline()`) строку нужно разбить на отдельные части по пробелам (каждая часть – символьная запись числа) и затем каждую часть преобразовать в целое число; например, чтение двух чисел:

```
s = Fin.readline()
```

```
symData = s.split()
```

```
x, y = map( int, symData )
```

или в компактной форме

В языке PascalABC.NET для чтения данных проще всего просто перенаправить входной поток на файл:

```
Assign( input, '26.txt' );
```

после этого можно использовать операторы `read` и `readln`, так же, как при вводе с клавиатуры

В языке C++ можно читать данные с помощью входного потока (`fstream`):

```
#include <fstream>
```

```
...
```

```
ifstream Fin("26.txt");
```

```
Fin >> x;
```

```
Fin >> y >> z;
```

РЕШЕНИЕ С ПОМОЩЬЮ ПРОГРАММЫ

Хранение массива данных

В языке PascalABC.NET используем динамический массив; когда станет известен его размер, выделаем место в памяти и читаем из входного потока:

```
var data: array of integer;  
SetLength( data, N );  
for var i:=0 to N-1 do  
    read( data[i] );
```

В языке Python для хранения массива данных используется список; следующая программа показывает чтение массива данных размера N в список **data** из файла «26.txt» (данные записаны в столбик):

```
data = [0]*N  
with open("26.txt") as Fin:  
    for i in range(N):  
        data[i] = int( Fin.readline() )
```

В языке C++ аналогично используется коллекция **vector**:

```
#include <vector>  
  
...  
vector <int> data(N);  
for( int i = 0; i < N; i++ )  
    Fin >> data[i];
```

ПРИМЕР

PascalABC.NET

```
var S, N: integer;
    data: array of integer;
begin
    Assign( input, '26.txt' );
    readln( S, N );
    SetLength( data, N );
    for var i:=0 to N-1 do
        read( data[i] );
    Sort( data );
    var total := 0;
    var count := 0;
    while count < N-1 do begin
        if total + data[count] > S then break;
        total += data[count];
        count += 1;
    end;
    var delta := S - total;
    var candidates := data.Where(
        x -> x - data[count-1] <= delta );
    Println( count, candidates.Max )
end.
```

Python

```
with open("26.txt") as Fin:
    data = Fin.readlines()
S, _ = map( int, data[0].split() )
del data[0]
data = sorted( list(map(int, data)) )

total = 0
for i, val in enumerate(data):
    if total + val > S: break
    total += val
print(i)

delta = S - total
candidates = [x for x in data
              if x-data[i-1] <= delta]
print( max(candidates) )
```

ЗАДАНИЕ 27 (ВЫСОКИЙ УРОВЕНЬ) ВРЕМЯ – 35 МИН)

27 (высокий уровень, время – 35 мин)

Тема: Обработка данных, вводимых из файла в виде последовательности чисел.

Что проверяется:

Умение создавать собственные программы (20–40 строк) для анализа числовых последовательностей.

1.6.3. Построение алгоритмов и практические вычисления.

1.1.5 Создавать программы на языке программирования по их описанию

Что нужно знать:

- как прочитать данные из файла
- основы комбинаторики
- динамическое программирование



Задание выполняется с использованием прилагаемых файлов.

27

Имеется набор данных, состоящий из пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 3 и при этом была максимально возможной. Гарантируется, что искомую сумму получить можно. Программа должна напечатать одно число – максимально возможную сумму, соответствующую условиям задачи.

Входные данные.

Даны два входных файла (файл A и файл B), каждый из которых содержит в первой строке количество пар N ($1 \leq N \leq 100000$). Каждая из следующих N строк содержит два натуральных числа, не превышающих 10 000.

Пример организации исходных данных во входном файле:

```
6
1 3
5 12
6 9
5 4
3 3
1 1
```

Для указанных входных данных значением искомой суммы должно быть число 32.

В ответе укажите два числа: сначала значение искомой суммы для файла A , затем для файла B .

Предупреждение: для обработки файла B не следует использовать переборный алгоритм, вычисляющий сумму для всех возможных вариантов, поскольку написанная по такому алгоритму программа будет выполняться слишком долго.

Ответ:

РЕШЕНИЕ С ПОМОЩЬЮ ПРОГРАММЫ

```
#include <iostream>
#include <fstream>
#include <cmath>
using namespace std;
int main() {
    ifstream Fin( "27-b.txt" );
    int N, s = 0, dMin = 10001;
    Fin >> N;
    for( int i = 0; i < N; i++ ) {
        int a, b;
        Fin >> a >> b;
        s += max( a, b ); // суммируем максимальные числа из каждой пары
        int d = abs( a-b ); // выбираем пару, в которой разница между числами минимальная
        и не делится на 3 (иначе делимость новой суммы на 3 сохранится)
        if( d % 3 > 0 )
            dMin = min( d, dMin );
    }
    if( s % 3 != 0 ) //если полученная сумма не делится на 3
        cout << s; //ТО ВЫВОДИМ s
    else
        cout << s-dMin; // иначе заменим число в одной из пар на второе, минимальное
}
```



Задание выполняется с использованием прилагаемых файлов.

27

Имеется набор данных, состоящий из пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 3 и при этом была максимально возможной. Гарантируется, что искомую сумму получить можно. Программа должна напечатать одно число – максимально возможную сумму, соответствующую условиям задачи.

Входные данные.

Даны два входных файла (файл *A* и файл *B*), каждый из которых содержит в первой строке количество пар N ($1 \leq N \leq 100000$). Каждая из следующих N строк содержит два натуральных числа, не превышающих 10 000.

Пример организации исходных данных во входном файле:

```
6
1 3
5 12
6 9
5 4
3 3
1 1
```

Для указанных входных данных значением искомой суммы должно быть число 32.

В ответе укажите два числа: сначала значение искомой суммы для файла *A*, затем для файла *B*.

Предупреждение: для обработки файла *B* не следует использовать переборный алгоритм, вычисляющий сумму для всех возможных вариантов, поскольку написанная по такому алгоритму программа будет выполняться слишком долго.

Ответ:

РЕШЕНИЕ С ПОМОЩЬЮ ЭЛЕКТРОННЫХ ТАБЛИЦ

- загружаем текст в Excel с помощью вкладки Данные/Из текста (используем разделитель - пробел); удаляем первую строку где только одно число (количество)

| | A | B | C |
|---|------|------|---|
| 1 | 5627 | 5841 | |
| 2 | 5544 | 6520 | |
| 3 | 1449 | 3580 | |
| 4 | 2984 | 5984 | |
| 5 | 6164 | 2583 | |
| 6 | 9588 | 3467 | |
| 7 | 1440 | 8636 | |

- в столбце C находим максимальное из каждой пары чисел: вставляем в C1 формулу =МАКС(A1:B1) и копируем ее вниз (двойной щелчок на маркере заполнения)
- вычисляем сумму чисел в столбце C
- проверяем кратность 3: находим остаток от деления на 3 по формуле =ОСТАТ(C21;3)
- если остаток не равен 0, то искомое число найдено; иначе продолжаем...
- в столбце D для каждой пары находим модуль разности по формуле вида =ABS(A1-B1);

- поскольку нам нужны только пары, в которых не делится на 3, записываем 0 в те строчки, где условие не выполняется; для этого используем условие:

=ЕСЛИ(ОСТАТ(A1-B1;3)=0;0;ABS(A1-B1))

| D1 | A | B | C | D | E | F | G |
|----|------|------|------|------|---|---|---|
| | 5627 | 5841 | 5841 | 214 | | | |
| | 5544 | 6520 | 6520 | 976 | | | |
| | 1449 | 3580 | 3580 | 2131 | | | |
| | 2984 | 5984 | 5984 | 0 | | | |

- сортируем диапазон по столбцу D по возрастанию и берем из него первое ненулевое число (минимальную разницу в паре)

| | A | B | C | D |
|---|------|------|------|-----|
| 1 | 2984 | 5984 | 5984 | 0 |
| 2 | 4221 | 5994 | 5994 | 0 |
| 3 | 6926 | 2390 | 6926 | 0 |
| 4 | 3702 | 6714 | 6714 | 0 |
| 5 | 2278 | 7180 | 7180 | 0 |
| 6 | 5627 | 5841 | 5841 | 214 |
| 7 | 7706 | 8023 | 8023 | 317 |

- вычитаем его из:

| | | | | |
|----|------|------|--------|---------|
| 17 | 9156 | 3466 | 9156 | 5690 |
| 18 | 9588 | 3467 | 9588 | 6121 |
| 19 | 2294 | 8733 | 8733 | 6439 |
| 20 | 1440 | 8636 | 8636 | 7196 |
| 21 | | | 127341 | =C21-D6 |

- можно заполнить столбец E формулами вида =\$C\$21-D1, тогда результат будет выведен сразу справа от минимальной разности

| E6 | A | B | C | D | E |
|----|------|------|------|-----|--------|
| | 2984 | 5984 | 5984 | 0 | 127341 |
| | 4221 | 5994 | 5994 | 0 | 127341 |
| | 6926 | 2390 | 6926 | 0 | 127341 |
| | 3702 | 6714 | 6714 | 0 | 127341 |
| | 2278 | 7180 | 7180 | 0 | 127341 |
| | 5627 | 5841 | 5841 | 214 | 127127 |
| | 7706 | 8023 | 8023 | 317 | 127024 |

- получаем 127127, для варианта B -

РЕСУРСЫ СЕТИ ИНТЕРНЕТ ДЛЯ САМООБРАЗОВАНИЯ УЧИТЕЛЯ ПО МЕТОДИКЕ ПРЕПОДАВАНИЯ ИНФОРМАТИКИ И ПОДГОТОВКЕ К СДАЧЕ ЕГЭ

Тренажёры для подготовки к компьютерному ЕГЭ (КЕГЭ)

1. [Тренажер для подготовки к КЕГЭ \(К. Поляков\)](#)
2. [Тренажер для подготовки к КЕГЭ \(А. Кабанов\)](#)

Для самообразования учителя по методике преподавания можно рекомендовать следующие ресурсы сети Интернет:

1. <http://www.fipi.ru> - Федеральный институт педагогических измерений
2. <https://inf-oge.sdamgia.ru/> - тренировочные тесты
3. https://neznaika.pro/oge/inf_oge/ - тренировочные тесты
4. <http://distan-school.ru/oge/?tap=3> - тренировочные тесты
5. <http://kpolyakov.spb.ru/school/oge.htm> - тренировочные тесты, в том числе тренировочные тесты для компьютерного ЕГЭ
6. <http://kpolyakov.spb.ru> – сайт учителя информатики К.Ю. Полякова

Материалы от Alex Danov:

Базовые алгоритмы для решения задач ЕГЭ на программирование:

https://www.youtube.com/playlist?list=PLXZ932--vml_-BWxVtEdU-p-BtnYfR8p

Решения задач из сборника 2020 г.: <https://github.com/AlexDanov/InfEGE-27-PascalABC>

Разборы задач 27 (из сборника 2020 г.):

https://www.youtube.com/playlist?list=PLXZ932--vml_ivq9QOC_gpZ2czSe2CLTU