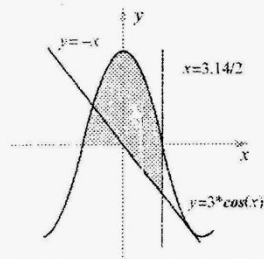


Часть 3

Для записи ответов к заданиям этой части (С1—С4) используйте бланк ответов № 2. Запишите сначала номер задания (С1, С2 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.

С1 Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y — действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.



ПАСКАЛЬ	БЕЙСИК	СИ
<pre>var x, y: real; begin readln(x, y); if y >= -x then if y >= 0 then if x <= 3.14/2 then if y <= 3*cos(x) then</pre>	<pre>INPUT x,y IF y >= -x THEN IF y >= 0 THEN IF x <= 3.14/2 THEN IF y <= 3*COS(x) THEN</pre>	<pre>void main(void) {float x,y; scanf("%f%f", &x, &y); if (y >= -x) if (y >= 0) if (x <= 3.14/2) if (y <= 3*cos(x))</pre>

Продолжение таблицы

<pre>write('принадлежит') else write('не принадлежит') end.</pre>	<pre>PRINT "принадлежит" ELSE PRINT "не принадле- жит" ENDIF ENDIF ENDIF ENDIF END</pre>	<pre>print("принадле- жит"); else print("не принадле- жит"); ;</pre>
---	--	--

Последовательно выполните следующее:

1. Приведите пример таких чисел x , y , при которых программа неверно решает поставленную задачу. Объясните, почему для указанных чисел программа неверно решает поставленную задачу.
2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

C2

Дан целочисленный массив из 65 элементов. Элементы массива могут принимать целые значения от -3000 до 3000 . Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести максимальное значение среди элементов массива, которые делятся на 3 или на 7. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого кратно 3 или 7.

Фрагмент программы с объявлением переменных и заполнением массива приведён ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Паскаль	Бейсик
<pre>Const N=65; var a: array [1..N] of integer; i, j, max: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre>N=65 DIM A(N) AS INTEGER DIM I, J, MAX AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
Си	Естественный язык
<pre>#include <stdio.h> #define N 65 void main(void){ int a[N]; int i, j, max; for (i=0; i<N; i++) scanf("% d", &a[i]); ... }</pre>	<p>Объявляем массив A из 65 элементов.</p> <p>Объявляем целочисленные переменные I, J, MAX. В цикле от 1 до 65 вводим элементы массива A с 1-го по 65-й.</p> <p>...</p>

В качестве ответа вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка про-

граммирования, например Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учётом синтаксиса и особенностей используемого вами языка программирования.

C3

У исполнителя Вычислитель две команды, которым присвоены номера:

1. прибавь 3
2. умножь на 2

Первая команда увеличивает число на 3, вторая умножает число на 2.

Программа для Вычислителя — это последовательность команд.

Сколько есть программ, которые число 2 преобразуют в число 29?

Ответ обоснуйте.

C4

Пользователю оператора голосовой Интернет-связи доступны не более 10 номеров, у каждого номера — уникальный логин. Система полностью доверяет клиентам, потому баланс каждого номера (целое число) может быть отрицательным. Каждому номеру соответствует отдельный счёт, хранящий информацию о балансе средств на нём. При использовании удалённого доступа к системе сервер регистрирует логин пользователя и величину изменения счёта. Предполагается, что начальный баланс каждого номера равен нулю. Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая будет составлять отчёт по хранимым пользовательским номерам и определять количество счетов, баланс которых в результате всех операций стал отрицательным.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

На вход программе в первой строке подаётся количество пришедших запросов на изменение счёта N . В каждой из последующих N строк записаны уникальный пользовательский логин и целое число — величина изменения, разделённые пробелом. Длина строки не превосходит 100 символов, логин пользователя может содержать буквы и цифры.

Пример входных данных:

```
5
234001 100
123000 -10
234001 -156
DD101101 123
123000 500
```

Программа должна вывести уникальные имена всех счетов, встречающихся в списке, в порядке убывания (не возрастания) количества средств на счёте. При этом каждый счёт должен быть выведен ровно один раз. После вывода баланса пользовательских счетов программа должна вывести количество счетов с отрицательным балансом.

Пример выходных данных для приведённого выше примера входных данных:

```
123000 490
DD101101 123
234001 -56
```

1

C1 1) Пример: $x = 1, y = -1$.

При ложном значении условия в любом из первых двух операторов ветвления программа не выводит никакого сообщения.

2) Возможная доработка -- заменить все операторы ветвления программы на язык Паскаль:

```
if (y>=0) and (x<=0) and (x>= -3.14/2) and (y<=3*cos(x)) or
   (x<=3.14/2) and (x>=0) and (y>=-x) and (y<=3*cos(x)) then
  writeln('принадлежит')
else
  writeln('не принадлежит');
```

C2 Фрагмент программы с решением на языке Паскаль:

```
max:= -3000;
{Перебираем элементы массива}
for i:=1 to n do
  if (a[i] mod 3 = 0) or (a[i] mod 7 = 0) then
    {Нашли элемент, который делится на 3 или на 7 }
    if max < a[i] then max:= a[i];
{Выводим максимальное значение из элементов массива, }
{которые делятся на 3 или на 7}
writeln(' ', max)
```

C3 Введём функцию $F(n)$, значением которой является количество программ Вычислителя, преобразующих число 2 к аргументу функции n . Следует определить, как вычислять функцию для чётных и нечётных аргументов, а также начальные значения функции.

1) Начальные значения функции. Выполним обе команды для исходного числа: Команда 1: $2 + 3 = 5$; Команда 2: $2 \cdot 2 = 4$. Никакими другими программами невозможно получить эти числа, следовательно, $F(4) = F(5) = 1$.

Все числа меньше 4 невозможно получить из 2, так как команды только увеличивают исходное число, значит, $F(n) = 0, n < 4$.

2) Нечётные числа $n = 2k - 1$ можно получить только одной командой -- прибавь 3, следовательно, количество программ получения нечётного числа и чётного $2k - 4$ одинаковое:

$$F(2k - 1) = F(2k - 1 - 3) = F(2k - 4).$$

3) Чётные числа можно получить как командой 1, так и командой 2: $F(2k) = F(2k) + F(2k - 3)$.

Таким образом, получены рекуррентные соотношения:

$$F(2k) = F(2k) + F(2k - 3), k > 2;$$

$$F(2k - 1) = F(2k - 4), k > 2;$$

$$F(4) = 1,$$

$$F(5) = 1,$$

$$F(n) = 0, n < 4;$$

по которым и будем проводить вычисления:

$$\begin{aligned} F(29) &= F(26) = F(23) + F(13) = F(20) + F(10) = F(17) + 2 \cdot F(10) = \\ &= F(14) + 2 \cdot F(10) = F(11) + 2 \cdot F(10) + F(7) = F(8) + 2 \cdot F(5) + 2 \cdot F(4) + F(4) = \\ &= F(5) + F(4) + 2 \cdot F(5) + 3 \cdot F(4) = 3 \cdot F(5) + 4 \cdot F(4) = 3 + 4 + 0 = 7 \end{aligned}$$

Ответ: 7

C4

Код программы на языке Borland Pascal 7.0

```
const NUM=10; {Максимальное количество номеров}
type account=record {Тип для представления одного номера}
  login:string[100]; {Логин номера}
  bal:integer; {Баланс номера}
end;
```

```

var tel:array[1..NUM]of account;(Массив всех учитываемых номеров)
    N,i,j,m,temp:integer;
{n - количество входных строк,
i,j - переменные для обеспечения циклов,
m - количество учтённых логинов,
temp - временная переменная для считывания величины изменения баланса)
ch:char;(символьная переменная для считывания)
s:string[100];(наименование логина)
b:boolean;(флаг наличия в массиве логина)
t:account;(вспомогательная переменная для сортировки)
begin
    for i:=1 to NUM do begin
        tel[i].login:='';
        tel[i].bal:=0;
    end;
readln(N);(считываем количество входных строк)
m:=0;(указываем программе, что на данный момент учтённых логинов нет)
for i:=1 to N do begin(запуск цикла обработки входных данных)
    read(ch);(считывание одного символа)
    while(ch<>' ')do begin(цикл чтения логина)
        s:=s+ch;
        (добавление к строковой переменной символа)
        read(ch);(считывание нового символа)
    end;
    readln(temp);(величина изменения счёта)
b:=false;(предполагаем, что подобного логина нет)
    for j:=1 to m do(Цикл поиска подобного логина)
        if(tel[j].login=s)then begin
(Если подобный логин обнаружен, изменение баланса счёта на величину temp)
            inc(tel[j].bal,temp);
            b:=true;
            (логин уже есть в массиве)
            break;
            (завершение поиска логина)
        end;
    if(not b)then begin
        (если логина ещё нет в массиве)
        inc(m);
        (увеличение количества хранимых логинов)
        tel[m].bal:=temp;
        (запись баланса temp)
        tel[m].login:=s;
        (запись логина)
    end;
    s:='';(Обнуляем строковую переменную)
end;
( сортируем записи в порядке убывания )
for i:=1 to m do
    for j:=1 to m-i do
        if(tel[j+1].bal> tel[j].bal) then begin
            t:=tel[j];
            tel[j]:= tel[j+1];
            tel[j+1]:=t;
        end;
j:=0; {Количество отрицательных балансов}
for i:=1 to m do begin
    writeln(tel[i].login,' ', tel[i].bal);
    (Вывод логина и баланса)
    if(tel[i].bal<0) then inc(j,1);
(если сумма на счёте меньше 0, увеличиваем количество счетов с отрицательным балансом)
end;
(вывод количества счетов с отрицательным балансом)
writeln(j);
end.

```