

5 вариант		6 вариант																																																																							
1.	30	1.	0																																																																						
2.	yzx	2.	xzy																																																																						
3.	13	3.	22																																																																						
4.	2876	4.	3																																																																						
5.	1	5.	115612																																																																						
6.	169	6.	12111																																																																						
7.	B5	7.	1																																																																						
8.	150	8.	99																																																																						
9.	28.4	9.	3750																																																																						
10.	ИИИЖЕ	10.	320																																																																						
11.	34	11.	16																																																																						
12.	29	12.	2046																																																																						
13.	64	13.	26																																																																						
14.	2	14.	20																																																																						
15.	42	15.	84																																																																						
16.	691012	16.	4																																																																						
17.	3412	17.	bcda																																																																						
18.	19	18.	20																																																																						
19.	8	19.	1798																																																																						
20.	130	20.	106																																																																						
21.	10	21.	20																																																																						
22.	18	22.	24																																																																						
23.	44	23.	512																																																																						
24.	<p>Решение использует запись программы на Паскале. Допускается использование программы на трёх других языках.</p> <p>1. Программа выведет два числа: 3 и 0.</p> <p>2. Программа напечатает правильный ответ, например, для последовательности 0 -1 -1 2</p> <p><i>Замечание для проверяющего. Программа будет работать верно, если в последовательности содержится 0 и при этом последнее число последовательности будет равно количеству неотрицательных чисел в последовательности.</i></p> <p>3. Первая ошибка. Неверная инициализация произведения. Строка с ошибкой: p:=0; Возможный вариант исправления: p:=1;</p> <p>4. Вторая ошибка. Вместо значения переменной count выводится значение переменной x: writeln(x); Возможный вариант исправления: writeln(count);</p>	24.	<p>Элементы ответа:</p> <table border="1"> <thead> <tr> <th>Область</th> <th>Условие 1 (y>=-1)</th> <th>Условие 2 (y<=x)</th> <th>Условие 3 (y<=0)</th> <th>Условие 4 (x*x+y*y<=1)</th> <th>Программа выведет</th> <th>Область обрабатывается верно</th> </tr> </thead> <tbody> <tr><td>A</td><td>да</td><td>нет</td><td>—</td><td>—</td><td>—</td><td>нет</td></tr> <tr><td>B</td><td>да</td><td>да</td><td>нет</td><td>—</td><td>—</td><td>нет</td></tr> <tr><td>C</td><td>да</td><td>нет</td><td>—</td><td>—</td><td>—</td><td>нет</td></tr> <tr><td>D</td><td>да</td><td>да</td><td>нет</td><td>—</td><td>—</td><td>нет</td></tr> <tr><td>E</td><td>да</td><td>нет</td><td>—</td><td>—</td><td>—</td><td>нет</td></tr> <tr><td>F</td><td>да</td><td>нет</td><td>—</td><td>—</td><td>—</td><td>нет</td></tr> <tr><td>G</td><td>да</td><td>да</td><td>да</td><td>да</td><td>принадлежит</td><td>да</td></tr> <tr><td>H</td><td>да</td><td>да</td><td>да</td><td>нет</td><td>не принадлежит</td><td>да</td></tr> <tr><td>I</td><td>нет</td><td>—</td><td>—</td><td>—</td><td>—</td><td>нет</td></tr> </tbody> </table>	Область	Условие 1 (y>=-1)	Условие 2 (y<=x)	Условие 3 (y<=0)	Условие 4 (x*x+y*y<=1)	Программа выведет	Область обрабатывается верно	A	да	нет	—	—	—	нет	B	да	да	нет	—	—	нет	C	да	нет	—	—	—	нет	D	да	да	нет	—	—	нет	E	да	нет	—	—	—	нет	F	да	нет	—	—	—	нет	G	да	да	да	да	принадлежит	да	H	да	да	да	нет	не принадлежит	да	I	нет	—	—	—	—	нет
Область	Условие 1 (y>=-1)	Условие 2 (y<=x)	Условие 3 (y<=0)	Условие 4 (x*x+y*y<=1)	Программа выведет	Область обрабатывается верно																																																																			
A	да	нет	—	—	—	нет																																																																			
B	да	да	нет	—	—	нет																																																																			
C	да	нет	—	—	—	нет																																																																			
D	да	да	нет	—	—	нет																																																																			
E	да	нет	—	—	—	нет																																																																			
F	да	нет	—	—	—	нет																																																																			
G	да	да	да	да	принадлежит	да																																																																			
H	да	да	да	нет	не принадлежит	да																																																																			
I	нет	—	—	—	—	нет																																																																			

J	да	да	да	нет	не принадлежит	нет
K	нет	—	—	—	—	нет

Возможная доработка (Паскаль, отдельно рассматриваем области, лежащие по разные стороны от прямой $x=0$):
`if (x*x+y*y<=1) and (y<=0) and (x>=0) or (x<=0) and (y>=-1) and (y<=x) then`
`write('принадлежит')`
`else write('не принадлежит')`

Возможная доработка (Си, отдельно рассматриваем области, лежащие по разные стороны от окружности):
`if((x*x+y*y<=1) && (y<=x) && (y<=0) || (x*x+y*y>=1) && (y<=x) && (y>=-1) && (x<=0))`
`cout << "принадлежит";`
`else cout << "не принадлежит";`

Обратите внимание! При разделении области вдоль какой-либо линии точки, которые лежат на этой линии внутри области, могут быть причислены к одной части, к другой или к обеим (то есть, например, в приведённом решении на языке Паскаль одно из условий $x>=0$ или $x<=0$) может быть строгим) Могут быть и другие верные способы доработки.

25. Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
 На языке Паскаль
`x:=0;`
`y:=0;`
`for i:=1 to N do`
`if (a[i] mod 2=1) then begin`
`x:=x+a[i];`
`y:=y+1;`
`end;`
`s:=x/y;`
`writeln(s);`

На языке Бейсик
`X = 0`
`Y = 0`
`FOR I = 1 TO N`
`IF A(I) MOD 2 = 1 THEN`
`X = X + A(I)`
`Y = Y + 1`
`ENDIF`
`NEXT I`
`S = X / Y`
`PRINT S`

На языке Си
`x=0;`
`y=0;`

25.	<p>ПАСКАЛЬ</p> <code>lmax := 0 ;</code> <code>l := 1;</code> <code>s := a[1];</code> <code>for i := 2 to N do</code> <code>if a[i] > a[i - 1] then</code> <code>begin</code> <code>l := l + 1;</code> <code>s := s + a[i];</code> <code>end</code> <code>else</code> <code>begin</code> <code>if l > lmax, then</code> <code>begin</code> <code>lmax := l;</code> <code>smax := s</code> <code>end;</code> <code>l := 1;</code> <code>s := a[i]</code> <code>end ;</code> <code>if l > lmax then</code> <code>smax := s;</code> <code>writeln(smax);</code> <p>СИ</p> <code>lmax = 0;</code> <code>l = 1;</code> <code>s = a [0] ;</code> <code>for(i =1; i < N; i++)</code> <code>if(a[i] > a[i - 1])</code> <code>{</code> <code>l++;</code>	<p>БЕЙСИК</p> <code>LMAX = 0</code> <code>L = 1</code> <code>S = A (1)</code> <code>FOR I = 2 TO N</code> <code>IF A (I) > A (I - 1) THEN</code> <code>L = L + 1</code> <code>S = S + A (I)</code> <code>ELSE</code> <code>IF L > LMAX THEN</code> <code>LMAX = L</code> <code>SMAX = S</code> <code>ENDIF</code> <code>L = 1</code> <code>S = A (I)</code> <code>ENDIF</code> <code>NEXT I</code> <code>IF L > LMAX THEN</code> <code>SMAX = S</code> <code>ENDIF</code> <code>PRINT SMAX</code> <p>Алгоритмический язык</p> <code>Lmax := 0</code> <code>L := 1</code> <code>S := a[i]</code> <code>нц для i от 2 до N</code> <code>если a[i] > a[i - 1]</code> <code>TO</code> <code>L := L + 1</code>
-----	--	--

```

for (i=0; i<N; i++)
if (a[i]%2==1)
{ x=x+a[i];
y++;
}
s=(float)x/y;
cout « s « endl

```

```

s += a[i];
}
else
{
if(l > lmax)
{
lmax = l;
smax = s;
}
l = 1;
s = a [ i ] ;
}
if(l > lmax)
smax = s;
cout « smax « endl;

```

```

S := S + a [ i ]
иначе
если L > Lmax
ТО
Lmax := L
Smax := S
все
L := 1
S := a[i]
все
кц
если L > Lmax
то
Smax := S
все
вывод Smax

```

26. Выигрывает Ваня.

Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделённые запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучах соответственно.

Стартовая позиция	1-й ход Петя (все варианты хода)	2-й ход Ваня (выигрышные ходы)	3-й ход Петя (все варианты хода, кроме непосредственно проигрышных)	4-й ход Ваня (выигрышные ходы)
2, 3	2, 9	2, 13	2, 17	2, 51
			6, 13	6, 39
	2, 7	2, 21	6, 21	6, 63
			2, 25	2, 75
	6, 3	18, 3	18, 9	54, 9
			22, 3	66, 3
		18, 7	54, 7	

Таблица содержит все возможные варианты ходов первого игрока. Из неё видно, что при любом ходе первого игрока у второго имеется ход, приводящий к победе.

26.

1а) Петя может выиграть, если $S = 13, \dots, 24$. Во всех этих случаях достаточно удвоить количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 24 камней.

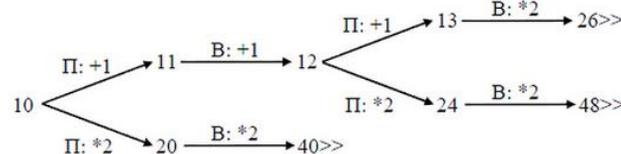
1б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 12$ камней. Тогда после первого хода Пети в куче будет 13 камней или 24 камня. В обоих случаях Ваня удваивает количество камней и выигрывает в один ход.

2. Возможные значения S : 6 и 11. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 12 камней: в первом случае удвоением, во втором добавлением одного камня. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выигрывает.

3. Возможное значение S : 10. После первого хода Пети в куче будет 11 или 20 камней, если в куче станет 20 камней, Ваня удвоит количество камней и выиграет первым ходом. Ситуация, когда в куче 11 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчеркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)
10	$10 + 1 = 11$	$11 + 1 = 12$	$12 + 1 = 13$	$13 * 2 = 26$
			$12 * 2 = 24$	$24 * 2 = 48$
	$10 * 2 = 20$	$20 * 2 = 40$		



<p>27. Программа должна верно читать входные данные, не запоминая их все, а сразу подсчитывая в массиве, хранящем 12 целых чисел, количество учащих в каждой из параллелей. Затем с использованием этого массива ищется параллель с максимальным числом учеников. За дополнительный просмотр этого массива распечатывается информация об искомым параллелях. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, параллель с максимальным количеством учеников единственна).</p> <p>Пример правильной и эффективной программы на языке Паскаль:</p> <pre> var pc:array[1..12] of integer; p : 1..12; class:string[3]; c:char; max, i, N:integer; begin readln(N) ; for i:=1 to 12 do pc[i]:=0; for i:=1 to N do begin repeat read(c) until c=' '; {считана фамилия} repeat read(c) until c=' '; {считано имя} readln(class); {определяем номер параллели} if length(class)=2 then p:=ord(class[1])-ord('0') else p:=(ord(class[1])-ord('0'))*10+ ord(class[2])-ord('0'); pc[p]:=pc[p]+1; {учитываем ученика этой параллели} end; max:=0; for i:=1 to 12 do if pc[i]>max then max:=pc[i]; writeln('Максимум учеников в параллели:',max); for i:=1 to 12 do if pc[i]=max then write(i, ' '); readln end. </pre>	<p>27.</p>
---	------------