

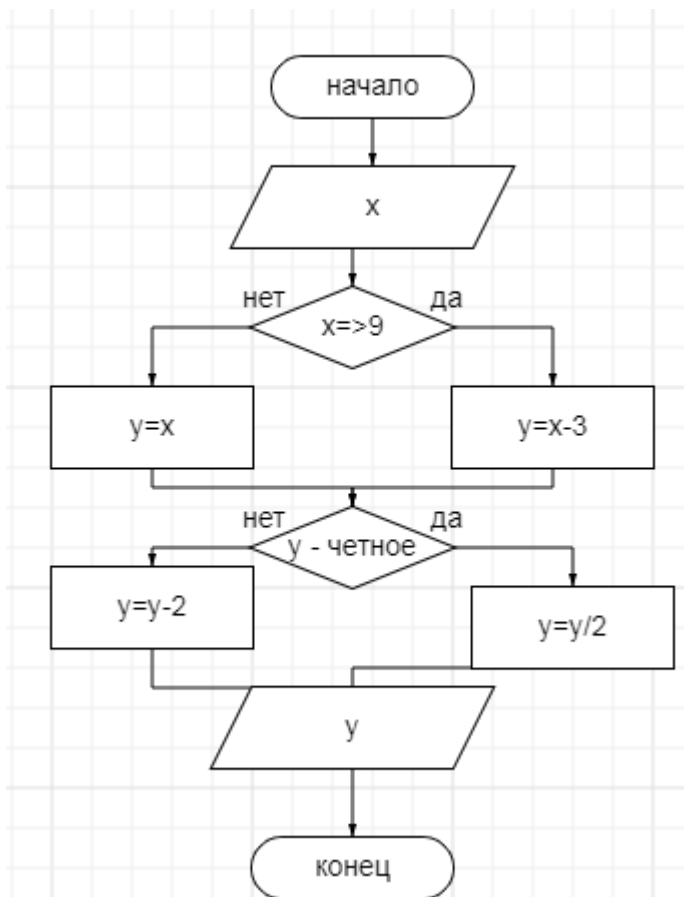


ИНФОРМАТИКА

ВАРИАНТ 1

Задача 1. (5 баллов)

Робот на вход подает число x , известно, что при выполнении условий на блок-схеме на выходе робот получил ответ $y=3$, укажите наименьшее положительное x , которое ввел робот. В решении докажите, что Ваш ответ верный.



Решение:

Сначала стоит описать алгоритм по блок-схеме:

1. Ввод x
2. Если $x \geq 0$, то $y = x - 3$, в противном случае $y = x$.
3. Если y – четное, то $y = y - 2$, в противном случае $y = y / 2$.
4. Вывод y .

Исходя из того, что в ответе Робот получил $y=3$, то на шаге 3 y было либо равно 6, либо 5. Далее рассматриваются варианты, если $y=6$, то был получен из разности $y=x-3$, тогда на вход подавалось число 9. Если рассмотреть вариант с пятеркой, тогда на вход должно было подаваться число 5. Доказано, что на вход должно подаваться число 5.



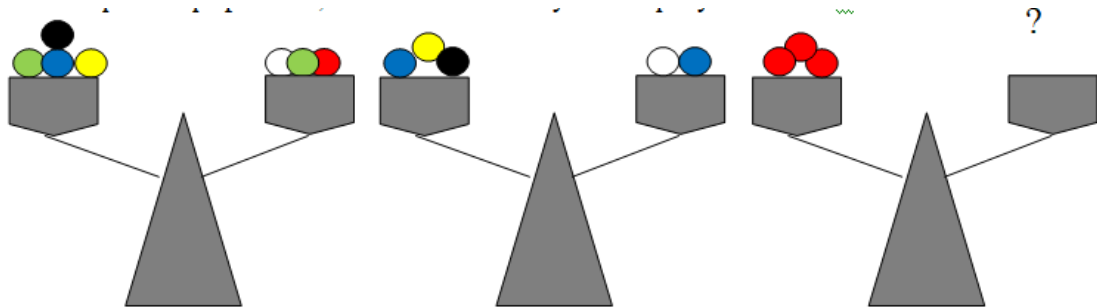
Ответ: 5.

Задача 2. (15 баллов)

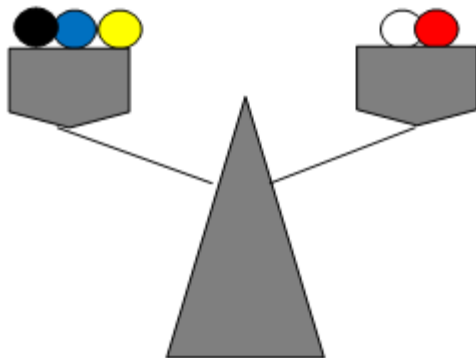
Гному известно, что зелёный, синий, желтый и черный камни весят вместе столько же сколько белый, зелёный и красный вместе. И что синий, жёлтый и чёрный камни вместе весят столько же, сколько и белый и синий вместе. Сколько будет весить 3 красных камня? Необходимо выразить вес 3 красных камней через веса зеленого, синего, жёлтого и черного камней. Можно выразить вес 3 красных камней через вес одного из вышеперечисленных камней. Камней одного цвет можно взять несколько. Решение поясните аналитически и графически.

Решение:

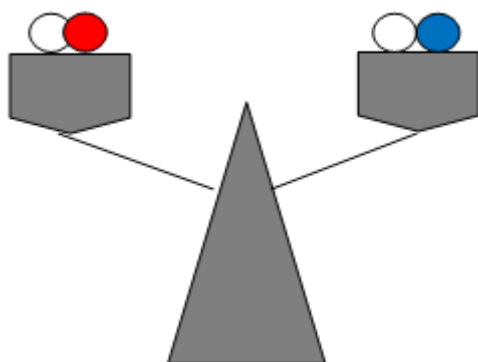
Изобразим графически, что известно гному и что требуется найти :



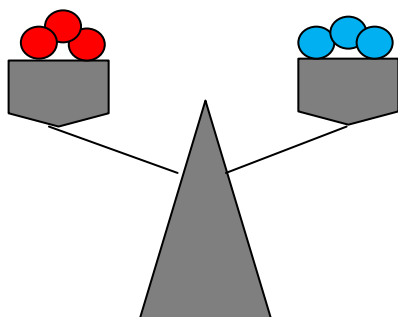
Если с первых весов и слева и справа убрать зелёный камень, то получим:



Получается, что на первых и вторых весах слева одни и те же камни: чёрный, синий и жёлтый. Поэтому на вторые веса слева положим камни с первых весов справа:



Теперь замечаем, что и слева лежит белый камень – уберем его. Получилось, что вес красного камня равен весу синего камня. Делаем вывод, что три красных камня весят столько же, сколько три синих камня





Задача 3. (10 баллов)

Дана система логических выражений:

$$\begin{cases} A \vee B \wedge \bar{C} \rightarrow 0 \\ \bar{C} \wedge B \vee A \wedge \bar{B} \rightarrow 0 \\ C \wedge A \vee \bar{B} \vee A \rightarrow 0 \end{cases}$$

Найти общий набор А, В, С, при которых системы обращаются в ложь. Ответ представить числом в десятичной системе счисления. (Например, набор 001 в десятичной системе 1 и т.д.)

Решение:

Чтобы решить данную задачу необходимо построить таблицы истинности для каждого из выражения, найти строки, в которых значение результирующего столбца будет соответствовать значению ложь, а потом сопоставить соответствующие наборы переменных.

Таблица истинности для первого выражения:

A	B	C	$\bar{C}$	$B \wedge \bar{C}$	$A \vee B \wedge \bar{C}$
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1

Таблица истинности для второго выражения:

A	B	C	$\bar{C}$	$\bar{B}$	$\bar{C} \wedge B$	$A \wedge \bar{B}$	$\bar{C} \wedge B \vee A \wedge \bar{B}$
0	0	0	1	1	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	0	0	0
1	0	0	1	1	0	1	1
1	0	1	0	1	0	1	1



1	1	0	1	0	1	0	1
1	1	1	0	0	0	0	0

Таблица истинности для третьего выражения:

A	B	C	$\bar{B}$	$C \wedge A$	$C \wedge A \vee \bar{B}$	$C \wedge A \vee \bar{B} \vee A$
0	0	0	1	0	1	1
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	1	1
1	0	1	1	1	1	1
1	1	0	0	0	0	1
1	1	1	0	1	1	1

Получается все три неравенства ложны, когда ABC соответствуют набор 011. Двоичное число 011 соответствует числу 3 в десятичной системе.

Ответ: 3.

Задача 4. (20 баллов)

Игорь попросил своих друзей помочь собрать ему яблоки и оставил для них коробку на сотню яблок. На следующий день Игорь забрал не полностью заполненную коробку с яблоками и попросил друзей написать ему о том, кто сколько собрал. Виктор написал, что собрал 23 яблока, Олег написал, что собрал 11, а Александр написал, что собрал 52. Игорь знает, что среди его друзей только он использует десятичную систему счисления, а остальные используют шестнадцатеричную, восьмеричную и двоичную, но не помнит кто какую. Сколько яблок насчитает в коробке Игорь?

Решение:

С учетом того, что Олег использует десятичную систему счисления, можно записать выражение:

$$M_{10} = 23_X + 11_Y + 52_Z < 100_{10},$$

где X, Y и Z основания систем счисления, которые могут быть равны 2, 8 и 16, а M – число яблок, которое насчитает Олег.



Из представленных слагаемых только число 11_7 может соответствовать двоичной системе счисления, поскольку цифры в записи чисел не могут быть больше основания системы счисления. Таким образом, Олег использует двоичную систему счисления и собрал 3_{10} яблока.

Если предположить, что Александр использует шестнадцатеричную систему счисления, а Виктор – восьмеричную, то количество яблок в коробке составит: $M_{10} = 19_{10} + 3_{10} + 82_{10} = 104_{10}$, что невозможно, так как в коробке не может быть больше 100 яблок.

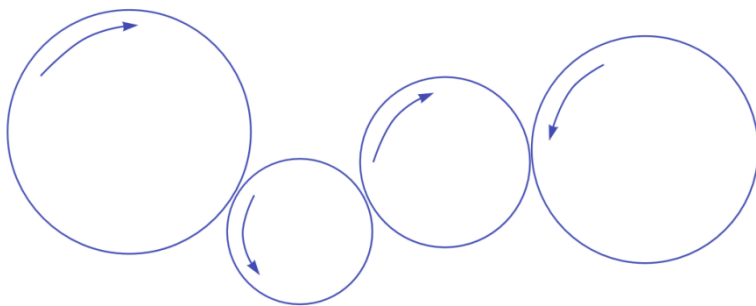
Если предположить, что Александр использует восьмеричную систему счисления, а Виктор – шестнадцатеричную, то количество яблок в коробке составит: $M_{10} = 35_{10} + 3_{10} + 42_{10} = 80_{10}$, что соответствует условию $M_{10} < 100_{10}$ и будет являться правильным ответом.

Ответ: 80

Задача 5. (25 баллов)

КОРОБКА ПЕРЕДАЧ

Однажды утром Андрей нашел в гараже отца N шестеренок разного радиуса от коробки передач старой машины. Он разложил их на столе так, чтобы каждая шестеренка (кроме первой и последней) касалась друг друга.



Он начал вращать первую шестеренку и заметил, что остальные тоже вращаются. Одни быстрее, другие медленнее. Обрадованный своим открытием, он решил посчитать, сколько раз повернутся остальные шестеренки, пока первая поворачивается один раз. Через некоторое время он сдался, заметив, что это число не всегда целое и не знает, что делать.

Напишите программу, которая определяет, сколько раз повернется каждая шестеренка за один оборот первой.



Входные данные: В первой строке целое число N – количество шестеренок, во второй радиусы шестеренок через пробел в том порядке, в котором Андрей разложил их на столе.

Результат должен содержать $N - 1$ строк, где в каждой строке печатается в консоль дробь A/B для каждой шестеренки, кроме первой, в том порядке, в котором они заданы. Дробь A/B означает, что соответствующая шестеренка поворачивается A/B раз, пока первая поворачивается один. Дробь должна быть в сокращенной форме (числитель и знаменатель не должны иметь общего делителя больше 1)

Несмотря на то, что в вашем задании указаны достаточно малые значения, ваша программа должна выполняться и для больших входных значений.

Пример:

Ввод: 3 8 4 2	Результат: 2/1 4/1
Ввод: 4 12 3 8 4	Результат: 4/1 3/2 3/1

Исходные данные:

5
10 50 12 65 10

Решение:

Количество оборотов i -го шестеренки зависит только от его радиуса и радиуса первой шестеренки. Искомое соотношение равно R_i / R_1 . Сократить дроби можно, разделив числитель и знаменатель на их наибольший общий делитель, заданный рекурсивной формулой:

$$GCD(a, b) = \begin{cases} a, & \text{где } b = 0 \\ GCD(b, a \bmod b), & \text{где } b \neq 0 \end{cases}$$



Решение на C++:

```
#include <stdio.h>

int gcd(int a, int b) { return b != 0 ? gcd(b, a % b) : a; }

int main(void) {
    int n, first, k;
    scanf("%d%d", &n, &first);
    for (int i = 0; i < n - 1; i++) {
        scanf("%d", &k);
        printf("%d/%d\n", first / gcd(first, k), k / gcd(first, k));
    }
    return 0;
}
```

Решение на Python:

```
def gcd(a, b):
    return gcd(b, a % b) if b != 0 else a

n = int(input())

shesterni = list(map(int, input().split()))
first = shesterni[0]
for k in shesterni[1:]:
    print(f"{first // gcd(first, k)}/{k // gcd(first, k)}")
```

Демонстрация работы кода C++



```
Granit2024 (Глобальная область)
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio>
4
5  int gcd(int a, int b) { return b != 0 ? gcd(b, a % b) : a; }
6
7  int main(void) {
8      int n, first, k;
9      scanf("%d%d", &n, &first);
10     for (int i = 0; i < n - 1; i++) {
11         scanf("%d", &k);
12         printf("%d/%d\n", first / gcd(first, k), k / gcd(first, k));
13     }
14     return 0;
15 }
16
17
18
```

Консоль отладки Microsoft V

3
8 4 2
2/1
4/1

Стр: 18 Симв: 1 Пробелы CRLF



```
Granit2024.cpp  X
Granit2024      (Глобальная область)
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio>
4
5  int gcd(int a, int b) { return b != 0 ? gcd(b, a % b) : a; }
6
7  int main(void) {
8      int n, first, k;
9      scanf("%d", &n, &first);
10     for (int i = 0; i < n - 1; i++) {
11         scanf("%d", &k);
12         printf("%d/%d\n", first / gcd(first, k), k / gcd(first, k));
13     }
14     return 0;
15 }
16
17
18
```

Консоль отладки Microsoft v

Вывод

4

12 3 8 4

4/1

3/2

3/1



```
Granit2024.cpp  X
Granit2024      (Глобальная область)
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio>
4
5  int gcd(int a, int b) { return b != 0 ? gcd(b, a % b) : a; }
6
7  int main(void) {
8      int n, first, k;
9      scanf("%d", &n, &first);
10     for (int i = 0; i < n - 1; i++) {
11         scanf("%d", &k);
12         printf("%d/%d\n", first / gcd(first, k), k / gcd(first, k));
13     }
14     return 0;
15 }
16
```

```
Консоль отладки Microsoft V  X
5
10 50 12 65 10
1/5
5/6
2/13
1/1
```

Ответ:

1/5

5/6

2/13

1/1

Задача 6. (25 баллов)

СЛОЖНОСТИ ОТНОШЕНИЙ

Однажды на контрольной работе по биологии Катя увидела, как её одноклассник Андрей случайно сломал манекен скелета Федора на уроке биологии. Катя девочка ответственная, к тому же староста класса, должна была рассказать об этом учителю, но поскольку её нравился Андрей она не стала этого делать. Вместо этого она предложила ему сходить вместе в кино, чтобы она «забыла» о случившемся.

Андрея не очень интересовали девочки в то время, потому что они отнимают у него драгоценное время, которое он обычно тратит на подготовку к олимпиаде по информатике. Поэтому он предложил ей сыграть в игру, и если Катя выиграет, то они вместе пойдут в кино. Катя согласилась, поскольку хорошо прыгает через скакалку и иногда даже гоняет футбольный мяч вместе с двумя своими братьями.



Андрей разложил на полу N листочков с целыми положительными числами по кругу и объяснил правила:

- Первый игрок берет любой листочек.
- Второй игрок берет любой из двух листочков, соседних с тем, которое взял первый игрок.
- Следующий игрок берет листочек, соседний с любым из взятых ранее листочков, и так далее, пока не кончатся листочки. Побеждает тот игрок, который возьмет больше листочков с нечетными числами (не кратных 2).

Андрей играет оптимально и всегда ищет стратегию, которая приведет к уверенной победе или ничьей. Андрей не знает, насколько хорошо играет Катя. Будучи настоящим кавалером, он позволил Кате сделать первый ход. Однако Катя думает только о том, чтобы сидеть рядом со Андреем перед большим экраном, поэтому она просит вас помощи в игре.

Напишите программу, которая определяет, сколько различных первых ходов может сделать Катя, чтобы у нее был шанс выиграть.

Входные данные: вам дается строка, которая содержит N ($1 \leq N \leq 100$) уникальных целых чисел, разделенных одиночными пробелами. Все числа будут находиться в диапазоне от 1 до 1000 (включительно).

Результат должен содержать одно целое число (количество первых ходов Кати)

Несмотря на то, что в вашем задании указаны достаточно малые значения, ваша программа должна выполняться и для больших входных значений.

Пример:

Ввод: 3 3 1 5	Результат: 3
Ввод: 4 1 2 3 4	Результат: 2



Ввод:	Результат:
8	5
4 10 5 2 9 8 1 7	

Пояснение:

Первый пример: Какое бы число Катя ни взяла первым, в итоге у нее будет два нечетных числа, а Андрей получит только одно.

Второй пример: Если Катя возьмет один из четных номеров, Андрей возьмет один из нечетных номеров и обеспечит ничью. Если она возьмет нечетное число, Андрей придется взять четное число и тогда проиграет, потому что Катя может взять другое нечетное число.

Исходные данные:

22

28 303 29 603 812 573 851 543 261 85 593 547 685 411 36 46 355 841 784 752 330 534

Решение:

В описании задачи стратегия Андрея прямо не указана, только то, что он старается выиграть, когда это возможно, т.е. ищет последовательность ходов, которая гарантирует победу. Одна из таких стратегий - попытаться выиграть с наибольшей разницей в счете.

Андрей не знает, насколько хорошо играет Катя, поэтому часть его стратегии заключается в том, чтобы предположить, что она тоже играет оптимально. так как он хочет быть абсолютно уверен, что не проиграет.

После первого хода Кати круг распадается, и мы остаемся с цепочкой чисел. Игроки поочередно берут числа с двух концов цепочки (они сами выбирают, с какого конца брать).

Предположим, что Катя сделала свой первый ход и осталась цепочка из $N-1$ чисел. Пусть $foo(a, b)$ - наибольшая разница в счете, которую может достичь игрок, находящийся в данный момент наверху (это число может быть отрицательным, если игрок всегда проигрывает), если в цепочке остались только числа, изначально имевшие индексы от a до b .



Текущий игрок либо берет число с индексом a и оставляет сопернику подзадачу $foo(a+1, b)$, либо берет число с индексом b и оставляет сопернику подзадачу $foo(a, b-1)$. Функция foo может быть эффективно вычислена для всех пар индексов с помощью динамического программирования.

Следующие тождества справедливы:

- $foo(k, k) = 1$, если число в индексе k нечетное, или 0 , если оно четное
- $foo(a, b) = \max(foo(a, a) - foo(a + 1, b), foo(b, b) - foo(a, b - 1))$

В решении Катя пробует каждое из N чисел в качестве первого хода, а затем использует функцию foo , чтобы вычислить, есть ли у нее выигрышная стратегия. Функция foo может быть вычислена сразу для всех цепочек, поскольку цепочки имеют много общих подзадач.

Реализация на C++

```
#include <stdio.h>
```

```
#define MAXN 100
```

```
int max( int a, int b ) { return a>b ? a : b; }
```

```
int main()
```

```
{
```

```
    int n, i, L;
```

```
    static int f[2*MAXN][2*MAXN];
```

```
    int rj;
```

```
    scanf( "%d", &n );
```

```
    for ( i=0; i<n; ++i ) {
```

```
        int x;
```

```
        scanf( "%d", &x );
```

```
        x %= 2;
```



```
f[i+n][i+n] = f[i][i] = x;
}

for ( L=2; L<=n; ++L )
    for ( i=0; i<2*n; ++i ) {
        int j = i+L-1;
        if ( j>=2*n ) break;

        f[i][j] = max( f[i][i] - f[i+1][j], f[j][j] - f[i][j-1] );
    }

    rj = 0;
    for ( i=0; i<n; ++i )
        if ( f[i][i] - f[i+1][i+n-1] > 0 )
            ++rj;
    printf( "%d\n", rj );

return 0;
}
```

Реализация на Python

```
def maximum(a, b):
    return a if a > b else b

n = int(input())
f = [[0] * (2 * n) for _ in range(2 * n)]

data = list(map(int, input().split()))
for i in range(n):
    x = data[i] % 2
    f[i + n][i + n] = f[i][i] = x

for L in range(2, n + 1):
```



```
for i in range(2 * n):  
    j = i + L - 1  
    if j >= 2 * n:  
        break  
    f[i][j] = maximum(f[i][i] - f[i + 1][j], f[j][j] - f[i][j - 1])
```

```
rj = 0  
for i in range(n):  
    if f[i][i] - f[i + 1][i + n - 1] > 0:  
        rj += 1
```

```
print(rj)
```

Демонстрация работы



```
Granit2024.cpp  X
Granit2024      (Глобальная область)  main()
13      int rj;
14
15      scanf("%d", &n);
16
17      for (i = 0; i < n; ++i) {
18          int x;
19          scanf("%d", &x);
20          x %= 2;
21          f[i + n][i + n] = f[i][i] = x;
22      }
23
24      for (L = 2; L <= n; ++L)
25          for (i = 0; i < 2 * n; ++i) {
26              int j = i + L - 1;
27              if (j >= 2 * n) break;
28
29              f[i][j] = max(f[i][i] - f[i + 1][j], f[j][j] - f[i][j - 1]);
30          }
31
32      rj = 0;
33      for (i = 0; i < n; ++i)
34          if (f[i][i] - f[i + 1][i + n - 1] > 0)
35              ++rj;
36      printf("%d\n", rj);
37
38      return 0;
39  }
40
41
42
```

Консоль отладки Microsoft V X + - □ X

80% 0 2 ↑ ↓ | Стр: 22 Симв: 6 Пробелы CRLF

4 10 5 2 9 8 1 7

5

Показать выходные данные из: Сборка



```
Granit2024.cpp  X
Granit2024      (Глобальная область)  main()
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4
5  #define MAXN 100
6
7  int max(int a, int b) { return a > b ? a : b; }
8
9  int main()
10 {
11     int n, i, L;
12     static int f[2 * MAXN][2 * MAXN];
13     int rj;
14
15     scanf("%d", &n);
16
17     for (i = 0; i < n; ++i) {
18         int x;
19         scanf("%d", &x);
20         x %= 2;
21         f[i + n][i + n] = f[i][i] = x;
22     }
23
24     for (L = 2; L <= n; ++L)
25     for (i = 0; i < 2 * n; ++i) {
26         int j = i + L - 1;
27         if (j >= 2 * n) break;
28
29         f[i][j] = max(f[i][i] - f[i + 1][j], f[j][j] - f[i][j - 1]);
30     }
31
32     rj = 0;
33     for (i = 0; i < n; ++i)
34         if (f[i][i] - f[i + 1][i + n - 1] > 0)
35             ++rj;
36     printf("%d\n", rj);
37 }
```

Консоль отладки Microsoft V X + -

22 % 0 2 Стр: 20 Симв: 16 Пробелы CRLF

28 303 29 603 812 573 851 543 261 85 593 547 685 411 36 46 355 841 784 752 330 534

19

Показать выходные данные из: Сборка

Ответ:

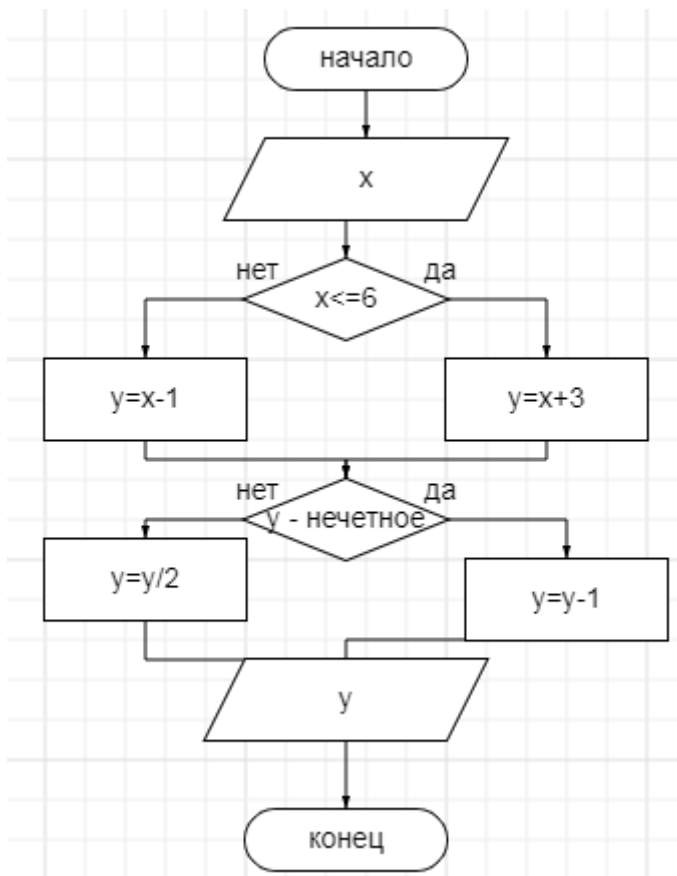
19



ИНФОРМАТИКА
ВАРИАНТ 2

Задача 1. (5 баллов)

Робот на вход подает число x , известно, что при выполнении условий на блок-схеме на выходе робот получил ответ $y=4$, укажите наименьшее положительное x , которое ввел робот. В решении докажите, что Ваш ответ верный.



Решение:

Сначала стоит описать алгоритм по блок-схеме:

1. Ввод x
2. Если $x \leq 6$, то $y = x + 3$, в противном случае $y = x - 1$.
3. Если y нечетное, то $y = y - 1$, в противном случае $y = y / 2$.
4. Вывод y .

Известно, что на выходе $y=4$. Поэтому y может быть равен либо 5 либо 8. Если $y=5$, то x был 2. Если $y=8$, то x был 5. 2 меньше 5, соответственно, ответ 2.

Ответ: 2.

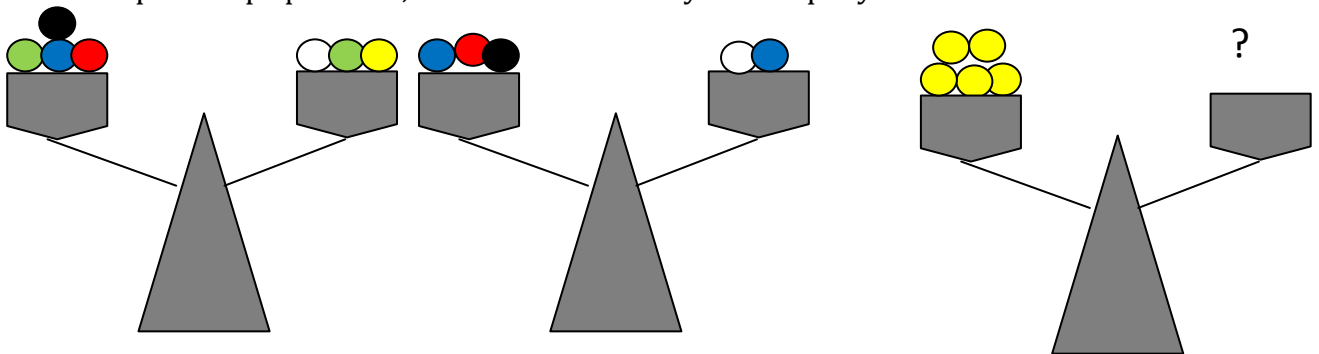


Задача 2. (15 баллов)

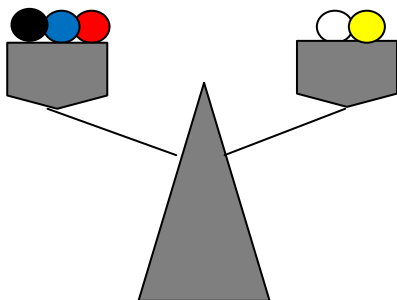
Гному известно, что зелёный, синий, красный и черный камни весят вместе столько же сколько белый, зелёный и желтый вместе. И что синий, красный и чёрный камни вместе весят столько же, сколько и белый и синий вместе. Сколько будет весить 5 жёлтых камней? Необходимо выразить вес 5 жёлтых камней через веса зеленого, синего, красного и черного камней. Можно выразить вес 5 жёлтых камней через вес одного из вышеперечисленных камней. Камней одного цвет можно взять несколько. Решение поясните аналитически и графически.

Решение:

Изобразим графически, что известно гному и что требуется найти :

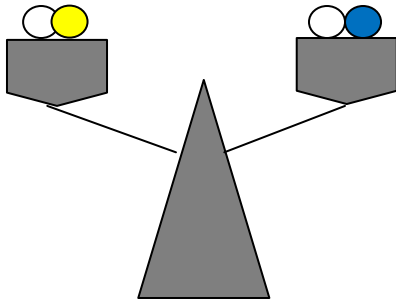


Если с первых весов и слева и справа убрать зелёный камень, то получим:

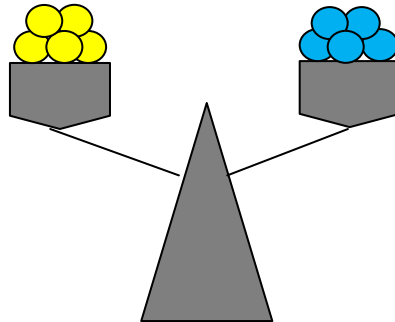




Получается, что на первых и вторых весах слева одни и те же камни: чёрный, синий и красный. Поэтому на вторые весы слева положим камни с первых весов справа:



Теперь замечаем, что и слева лежит белый камень – уберем его. Получилось, что вес жёлтого камня равен весу синего камня. Делаем вывод, что пять жёлтых камней весят столько же, сколько пять синих камней.



Задача 3.

Дана система логических выражений:

$$\begin{cases} A \vee \bar{B} \wedge C \vee A \rightarrow 0 \\ \bar{B} \wedge C \vee A \wedge C \rightarrow 0 \\ \bar{C} \wedge A \wedge B \vee C \rightarrow 0 \end{cases}$$

Найти общий набор A, B, C , при которых системы обращаются в ложь. Ответ представить числом в десятичной системе счисления. (Например, набор 001 в десятичной системе 1 и т.д.)

Решение:



Чтобы решить данную задачу необходимо построить таблицы истинности для каждого из выражения, найти строки, в которых значение результирующего столбца будет соответствовать значению ложь, а потом сопоставить соответствующие наборы переменных.

Таблица истинности для первого выражения:

A	B	C	$\bar{B}$	$\bar{B} \wedge C$	$A \vee \bar{B} \wedge C$	$A \vee \bar{B} \wedge C \vee A$
0	0	0	1	0	0	0
0	0	1	1	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	0	0	1	1
1	1	1	0	1	1	1

Таблица истинности для второго выражения:

A	B	C	$\bar{B}$	$\bar{B} \wedge C$	$A \wedge C$	$\bar{B} \wedge C \vee A \wedge C$
0	0	0	1	1	0	0
0	0	1	1	1	0	1
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	1	1	1	1
1	1	0	0	0	0	0
1	1	1	0	0	1	1

Таблица истинности для третьего выражения:

A	B	C	$\bar{C}$	$\bar{C} \wedge A$	$\bar{C} \wedge A \wedge B$	$\bar{C} \wedge A \wedge B \vee C$
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	0	1



1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	1	0	1	1	1	1
1	1	1	0	0	0	1

Получается все три неравенства ложны, когда ABC соответствуют набор 010. Двоичное число 010 соответствует числу 2 в десятичной системе.

Ответ: 2.

Задача 4. (20 баллов)

Игорь попросил своих друзей помочь собрать ему яблоки и оставил для них коробку. На следующий день Игорь насчитал в коробке 90 яблок и, для того, чтобы понять, кто из его друзей собрал больше, попросил каждого написать точное количество. Виктор написал, что собрал 30 яблок, Олег написал, что собрал 10, а Александр написал, что собрал 50. Игорь знал, что среди его друзей только он использует десятичную систему счисления, а остальные используют шестнадцатеричную, восьмеричную и двоичную, но не смог вспомнить кто какую. Тем не менее он смог разобраться и составил табличку с результатами. Сколько яблок будет записано в этой табличке у того, кто собрал больше всех?

Решение:

С учетом того, что Олег использует десятичную систему счисления, можно записать выражение:

$$90_{10} = 30_X + 10_Y + 50_Z,$$

где X, Y и Z основания систем счисления, которые могут быть равны 2, 8 и 16, а M – число яблок, которое насчитает Олег.

Из представленных слагаемых только число 10_Y может соответствовать двоичной системе счисления, поскольку цифры в записи чисел не могут быть больше основания системы счисления. Таким образом, Олег использует двоичную систему счисления и собрал 2_{10} яблока.

Если предположить, что Александр использует шестнадцатеричную систему счисления, а Виктор – восьмеричную, то количество яблок в коробке составит: $24_{10} + 2_{10} + 80_{10} = 106_{10}$, что невозможно, так как Игорь насчитал в коробке 90 яблок.

Если предположить, что Александр использует восьмеричную систему счисления, а Виктор – шестнадцатеричную, то количество яблок в коробке составит: $M_{10} = 48_{10} +$



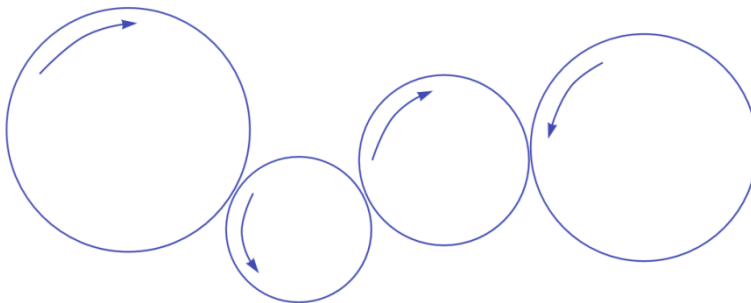
$2_{10} + 40_{10} = 90_{10}$, что соответствует условию, а количество яблок в десятичной системе счисления, равное 48, будет являться правильным ответом.

Ответ: 48

Задача 5. (25 баллов)

КОРОБКА ПЕРЕДАЧ

Однажды утром Андрей нашел в гараже отца N шестеренок разного радиуса от коробки передач старой машины. Он разложил их на столе так, чтобы каждая шестеренка (кроме первой и последней) касалась друг друга.



Он начал вращать первую шестеренку и заметил, что остальные тоже вращаются. Одни быстрее, другие медленнее. Обрадованный своим открытием, он решил посчитать, сколько раз повернутся остальные шестеренки, пока первая поворачивается один раз. Через некоторое время он сдался, заметив, что это число не всегда целое и не знает, что делать.

Напишите программу, которая определяет, сколько раз повернется каждая шестеренка за один оборот первой.

Входные данные: В первой строке целое число N – количество шестеренок, во второй радиусы шестеренок через пробел в том порядке, в котором Андрей разложил их на столе.

Результат должен содержать $N - 1$ строк, где в каждой строке печатается в консоль дробь A/B для каждой шестеренки, кроме первой, в том порядке, в котором они заданы. Дробь A/B означает, что соответствующая шестеренка поворачивается A/B раз, пока первая поворачивается один. Дробь должна быть в сокращенной форме (числитель и знаменатель не должны иметь общего делителя больше 1)



Несмотря на то, что в вашем задании указаны достаточно малые значения, ваша программа должна выполняться и для больших входных значений.

Пример:

Ввод:	Результат:
3	2/1
8 4 2	4/1
Ввод:	Результат:
4	4/1
12 3 8 4	3/2
	3/1

Исходные данные:

7
5 50 11 35 3 65 7

Решение:

Количество оборотов i -го шестеренки зависит только от его радиуса и радиуса первой шестеренки. Искомое соотношение равно R_i / R_1 . Сократить дроби можно, разделив числитель и знаменатель на их наибольший общий делитель, заданный рекурсивной формулой:

$$GCD(a, b) = \begin{cases} a, & \text{где } b = 0 \\ GCD(b, a \bmod b), & \text{где } b \neq 0 \end{cases}$$

Решение на C++:

```
#include <stdio.h>
```

```
int gcd(int a, int b) { return b != 0 ? gcd(b, a % b) : a; }
```

```
int main(void) {  
    int n, first, k;  
    scanf("%d%d", &n, &first);
```



```
for (int i = 0; i < n - 1; i++) {  
    scanf("%d", &k);  
    printf("%d/%d\n", first / gcd(first, k), k / gcd(first, k));  
}  
return 0;  
}
```

Решение на Python:

```
def gcd(a, b):  
    return gcd(b, a % b) if b != 0 else a  
  
n = int(input())  
  
shesterni = list(map(int, input().split()))  
first = shesterni[0]  
for k in shesterni[1:]:  
    print(f"{first // gcd(first, k)}/{k // gcd(first, k)}")
```

Демонстрация работы кода C++



```
Granit2024 (Глобальная область)
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4
5  int gcd(int a, int b) { return b != 0 ? gcd(b, a % b) : a; }
6
7  int main(void) {
8      int n, first, k;
9      scanf("%d%d", &n, &first);
10     for (int i = 0; i < n - 1; i++) {
11         scanf("%d", &k);
12         printf("%d/%d\n", first / gcd(first, k), k / gcd(first, k));
13     }
14     return 0;
15 }
16
17
18
```

Консоль отладки Microsoft V

3
8 4 2
2/1
4/1

Стр: 18 Симв: 1 Пробелы CRLF



```
Granit2024.cpp  X
Granit2024      (Глобальная область)
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio>
4
5  int gcd(int a, int b) { return b != 0 ? gcd(b, a % b) : a; }
6
7  int main(void) {
8      int n, first, k;
9      scanf("%d", &n, &first);
10     for (int i = 0; i < n - 1; i++) {
11         scanf("%d", &k);
12         printf("%d/%d\n", first / gcd(first, k), k / gcd(first, k));
13     }
14     return 0;
15 }
16
17
18
```

Консоль отладки Microsoft v

Вывод

4

12 3 8 4

4/1

3/2

3/1



```
Granit2024 (Глобальная область)
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio>
4
5  int gcd(int a, int b) { return b != 0 ? gcd(b, a % b) : a; }
6
7  int main(void) {
8      int n, first, k;
9      scanf("%d", &n, &first);
10     for (int i = 0; i < n - 1; i++) {
11         scanf("%d", &k);
12         printf("%d/%d\n", first / gcd(first, k), k / gcd(first, k));
13     }
14     return 0;
15 }
16
17
```

Консоль отладки Microsoft V

```
7
5 50 11 35 3 65 7
1/10
5/11
1/7
5/3
1/13
5/7
```

Ответ:

1/10

5/11

1/7

5/3

1/13

5/7

Задача 6.

СЛОЖНОСТИ ОТНОШЕНИЙ

Однажды на контрольной работе по биологии Катя увидела, как её одноклассник Андрей случайно сломал манекен скелета Федора на уроке биологии. Катя девочка ответственная, к тому же староста класса, должна была рассказать об этом учителю, но поскольку её нравился Андрей она не стала этого делать. Вместо этого она предложила ему сходить вместе в кино, чтобы она «забыла» о случившемся.

Андрея не очень интересовали девочки в то время, потому что они отнимают у него драгоценное время, которое он обычно тратит на подготовку к олимпиаде по информатике. Поэтому он предложил ей сыграть в игру, и если Катя выиграет, то



они вместе пойдут в кино. Катя согласилась, поскольку хорошо прыгает через скакалку и иногда даже гоняет футбольный мяч вместе с двумя своими братьями.

Андрей разложил на полу N листочков с целыми положительными числами по кругу и объяснил правила:

- Первый игрок берет любой листочек.
- Второй игрок берет любой из двух листочков, соседних с тем, которое взял первый игрок.
- Следующий игрок берет листочек, соседний с любым из взятых ранее листочков, и так далее, пока не кончатся листочки. Побеждает тот игрок, который возьмет больше листочков с нечетными числами (не кратных 2).

Андрей играет оптимально и всегда ищет стратегию, которая приведет к уверенной победе или ничьей. Андрей не знает, насколько хорошо играет Катя. Будучи настоящим кавалером, он позволил Кате сделать первый ход. Однако Катя думает только о том, чтобы сидеть рядом со Андреем перед большим экраном, поэтому она просит вас помощи в игре.

Напишите программу, которая определяет, сколько различных первых ходов может сделать Катя, чтобы у нее был шанс выиграть.

Входные данные: вам дается строка, которая содержит N ($1 \leq N \leq 100$) уникальных целых чисел, разделенных одиночными пробелами. Все числа будут находиться в диапазоне от 1 до 1000 (включительно).

Результат должен содержать одно целое число (количество первых ходов Кати)

Несмотря на то, что в вашем задании указаны достаточно малые значения, ваша программа должна выполняться и для больших входных значений.

Пример:

Ввод: 3 3 1 5	Результат: 3
Ввод: 4 1 2 3 4	Результат: 2



Ввод:	Результат:
8	5
4 10 5 2 9 8 1 7	

Пояснение:

Первый пример: Какое бы число Катя ни взяла первым, в итоге у нее будет два нечетных числа, а Андрей получит только одно.

Второй пример: Если Катя возьмет один из четных номеров, Андрей возьмет один из нечетных номеров и обеспечит ничью. Если она возьмет нечетное число, Андрей придется взять четное число и тогда проиграет, потому что Катя может взять другое нечетное число.

Исходные данные:

23

101 178 272 255 611 517 833 613 225 203 467 939 906 807 349 26 431 591 605 185 411
393 147

Решение:

В описании задачи стратегия Андрея прямо не указана, только то, что он старается выиграть, когда это возможно, т.е. ищет последовательность ходов, которая гарантирует победу. Одна из таких стратегий - попытаться выиграть с наибольшей разницей в счете.

Андрей не знает, насколько хорошо играет Катя, поэтому часть его стратегии заключается в том, чтобы предположить, что она тоже играет оптимально. так как он хочет быть абсолютно уверен, что не проиграет.

После первого хода Кати круг распадается, и мы остаемся с цепочкой чисел. Игроки поочередно берут числа с двух концов цепочки (они сами выбирают, с какого конца брать).

Предположим, что Катя сделала свой первый ход и осталась цепочка из $N-1$ чисел. Пусть $foo(a, b)$ - наибольшая разница в счете, которую может достичь игрок, находящийся в данный момент наверху (это число может быть отрицательным, если игрок всегда проигрывает), если в цепочке остались только числа, изначально имевшие индексы от a до b .



Текущий игрок либо берет число с индексом a и оставляет сопернику подзадачу $foo(a+1, b)$, либо берет число с индексом b и оставляет сопернику подзадачу $foo(a, b-1)$. Функция foo может быть эффективно вычислена для всех пар индексов с помощью динамического программирования.

Следующие тождества справедливы:

- $foo(k, k) = 1$, если число в индексе k нечетное, или 0 , если оно четное
- $foo(a, b) = \max(foo(a, a) - foo(a + 1, b), foo(b, b) - foo(a, b - 1))$

В решении Катя пробует каждое из N чисел в качестве первого хода, а затем использует функцию foo , чтобы вычислить, есть ли у нее выигрышная стратегия. Функция foo может быть вычислена сразу для всех цепочек, поскольку цепочки имеют много общих подзадач.

Реализация на C++

```
#include <stdio.h>
```

```
#define MAXN 100
```

```
int max( int a, int b ) { return a>b ? a : b; }
```

```
int main()
```

```
{
```

```
    int n, i, L;
```

```
    static int f[2*MAXN][2*MAXN];
```

```
    int rj;
```

```
    scanf( "%d", &n );
```

```
    for ( i=0; i<n; ++i ) {
```

```
        int x;
```

```
        scanf( "%d", &x );
```

```
        x %= 2;
```



```
f[i+n][i+n] = f[i][i] = x;
}

for ( L=2; L<=n; ++L )
    for ( i=0; i<2*n; ++i ) {
        int j = i+L-1;
        if ( j>=2*n ) break;

        f[i][j] = max( f[i][i] - f[i+1][j], f[j][j] - f[i][j-1] );
    }

    rj = 0;
    for ( i=0; i<n; ++i )
        if ( f[i][i] - f[i+1][i+n-1] > 0 )
            ++rj;
    printf( "%d\n", rj );

return 0;
}
```

Реализация на Python

```
def maximum(a, b):
    return a if a > b else b

n = int(input())
f = [[0] * (2 * n) for _ in range(2 * n)]

data = list(map(int, input().split()))
for i in range(n):
    x = data[i] % 2
```



$f[i + n][i + n] = f[i][i] = x$

```
for L in range(2, n + 1):
    for i in range(2 * n):
        j = i + L - 1
        if j >= 2 * n:
            break
        f[i][j] = maximum(f[i][i] - f[i + 1][j], f[j][j] - f[i][j - 1])

rj = 0
for i in range(n):
    if f[i][i] - f[i + 1][i + n - 1] > 0:
        rj += 1

print(rj)
```

Демонстрация работы



```
Granit2024.cpp  X
Granit2024      (Глобальная область)  main()
13      int rj;
14
15      scanf("%d", &n);
16
17      for (i = 0; i < n; ++i) {
18          int x;
19          scanf("%d", &x);
20          x %= 2;
21          f[i + n][i + n] = f[i][i] = x;
22      }
23
24      for (L = 2; L <= n; ++L)
25          for (i = 0; i < 2 * n; ++i) {
26              int j = i + L - 1;
27              if (j >= 2 * n) break;
28
29              f[i][j] = max(f[i][i] - f[i + 1][j], f[j][j] - f[i][j - 1]);
30          }
31
32      rj = 0;
33      for (i = 0; i < n; ++i)
34          if (f[i][i] - f[i + 1][i + n - 1] > 0)
35              ++rj;
36      printf("%d\n", rj);
37
38      return 0;
39  }
40
41
42
```

Консоль отладки Microsoft V X + - □ X

80% 0 2 ↑ ↓ | Стр: 22 Симв: 6 Пробелы CRLF

4 10 5 2 9 8 1 7

5

Показать выходные данные из: Сборка



```
Granit2024.cpp  X
Granit2024 (Глобальная область)
13     int rj;
14
15     scanf("%d", &n);
16
17     for (i = 0; i < n; ++i) {
18         int x;
19         scanf("%d", &x);
20         x %= 2;
21         f[i + n][i + n] = f[i][i] = x;
22     }
23
24     for (L = 2; L <= n; ++L)
25         for (i = 0; i < 2 * n; ++i) {
26             int j = i + L - 1;
27             if (j >= 2 * n) break;
28
29             f[i][j] = max(f[i][i] - f[i + 1][j], f[j][j] - f[i][j - 1]);
30         }
31
32     rj = 0;
33     for (i = 0; i < n; ++i)
34         if (f[i][i] - f[i + 1][i + n - 1] > 0)
35             ++rj;
36     printf("%d\n", rj);
37
38     return 0;
39 }
40
41
42
```

Консоль отладки Microsoft V X

23% 0 2 | Стр: 40 Симв: 1 Пробелы CRLF

101 178 272 255 611 517 833 613 225 203 467 939 906 807 349 26 431 591 605 185 411 393 147

9

Вывести выходные данные из: Сборка

Ответ:

9



Олимпиада школьников «Гранит науки» - 2024
