Задача A. Лето и шоколадки

1 июня — это «32 мая». Поэтому ответом будет $32 - n$.

Решение на Python:

```
print(32 - int(input()))
```

Задача B. Мультик

Заметим, что время просмотра никак не зависит от количества серий в одном сезоне. Важно только, сколько серий всего в мультсериале. Таким образом, введем $D = A + B$.

Далее нужно понять, сколько раз Сева пойдет наливать себе чай. Понятно, что эта величина напрямую зависит от количества «троек серий» — числа $\lfloor \frac{D}{3} \rfloor$ ($D/3$, округленного вниз), однако есть несколько нюансов. Во-первых, одну чашку Сева всегда нальет до просмотра. Во-вторых, он не будет наливать чашку после последней «тройки», если чашка опустела ровно на последней серии всего мультсериала, что бывает только, если D делится на 3.

Таким образом, если D не делится на 3, количество раз, сколько Сева нальет себе чай, равно $\lfloor \frac{D}{3} \rfloor + 1$, а если делится, то $\lfloor \frac{D}{3} \rfloor$. Остается только домножить это число на C, чтобы получить суммарное время на заваривание чая, и прибавить к нему $20 \cdot D$ — время на просмотр сериала.

Решение на Python:

```
a = int(input())
b = int(input())
c = int(input())

d = a + b
if d % 3 == 0:
     tea = d // 3
     print(tea * c + 20 * d)
else:
     tea = 1 + d // 3
     print(tea * c + 20 * d)
```

Задача C. Карточки с цифрами

Заметим, что для заданной перестановки карточек a_1, a_2, a_3, a_4 сумма будет $10 \cdot (a_1 + a_3) + (a_2 + a_4)$, откуда заметим, что менять имеет смысл только 1-ю и 2-ю, 1-ю и 4-ю, 3-ю и 2-ю или 3-ю и 4-ю карточки местами. Причем так как значение первой и третьей карточки умножатся на 10, имеет смысл сделать их как можно больше. Таким образом, достаточно просто проверить, можно ли увеличить цифру на первой или третьей карточке одним из четырех обменов.

Решение на Python:

```
a1 = int(input())
a2 = int(input())
a3 = int(input())
a4 = int(input())

if a1 < a2 or a1 < a4 or a3 < a2 or a3 < a4:
     print("YES")
else:
     print("NO")
```

Задача D. НОК

В этой задаче достаточно было просто перебрать результат при помощи цикла.

Решение на Python:

```
a = int(input())
```

```
b = int(input())
c = int(input())

ans = 0
for i in range(1, c + 1):
    if i % a == 0 and i % b == 0:
        ans = i
print(ans)
```

Задача E. Циклопы

В этой задаче были небольшие ограничения на длину строки, из-за чего можно было просто перебрать размер наибольшего циклопа и посмотреть, сколько таких входит в s.

Решение на Python:

```
n = int(input())
s = input()

max_cyclope = 100
while s.count('0' + ')' * max_cyclope) == 0:
    max_cyclope -= 1
print(s.count('0' + ')' * max_cyclope))
```

Задача F. Массив

В этой задаче нельзя было сымитировать процесс, поскольку поскольку n достаточно большое. Вместо этого попробуем решить эту задачу даже без создания массива.

Для этого предположим, что ответом является число k. Тогда для любого индекса i, что $1 \leqslant i < n$ верно, что $a_i + i \cdot k < a_{i+1} + (i+1) \cdot k$, что эквивалентно $a_i - a_{i+1} < k$. Так как k и $a_i - a_{i+1}$ — целые числа, то это выражение эквивалентно $a_i - a_{i+1} + 1 \leqslant k$

Таким образом, в качестве k следует выбрать наибольшее выражение $a_i - a_{i+1} + 1$ среди всех i или 0, если массив уже изначально возрастающий.

Решение на Python:

```
n = int(input())
prev = -1
k = 0
for i in range(n):
    c = int(input())
    if prev != -1:
        k = max(k, prev - c + 1)
    prev = c
print(k)
```

Задача G. Кубики сахара

Сначала посмотрим, каким образом считалось b. Вывести эту формулу несложно: $b = \lceil \frac{n}{c} \rceil \cdot k$.

Так как нас интересует c, для начала перепишем ее следующим образом: $\frac{b}{k} = \lceil \frac{n}{c} \rceil$. Заметим, что результат округления — целое число, поэтому слева тоже должно стоять целое число. Значит, если b не делится на k, то следует вывести 0. Иначе введем $d = \frac{b}{k}$.

Теперь мы понимаем, что просто так выразить c из $d = \lceil \frac{n}{c} \rceil$ нельзя. Однако давайте проанализируем, чему могла бы равняться дробь под округлением вверх. С одной стороны, она не больше d, но с другой — строго больше $d - 1$. Таким образом, мы имеем систему из двух неравенств:

$$\begin{cases} \frac{n}{c} \leqslant d \\ d - 1 < \frac{n}{c} \end{cases}$$

Если $d = 1$, то нетрудно показать, что c можно выбрать сколь угодно большим, поэтому ответ: -1. Однако, если это не так, то можно поменять выражение с d и знаменатель дроби местами. Получится следующая система:

$$\begin{cases} \frac{n}{d} \leqslant c \\ c < \frac{n}{d-1} \end{cases}$$

Значит, c может принимать любые целые значения в полуинтервале $[\frac{n}{d}; \frac{n}{d-1})$. Пусть $a = \lceil \frac{n}{d} \rceil$, а $b = \lceil \frac{n}{d-1} \rceil - 1$. Нетрудно заметить, что тогда c может принимать как раз значения $a, a+1, \ldots, b$.

Осталось только быстро посчитать сумму всех возможных вариантов. Это сделается при помощи формулы суммы арифметической прогрессии. Ответом будет $\frac{(b-a+1)\cdot(a+b)}{2}$.

Решение на Python:

```
from math import ceil
n = int(input())
k = int(input())
b = int(input())

if b % k != 0:
    print(0)
    exit(0)

d = b // k

if d == 1:
    print(-1)
    exit(0)

a = ceil(n / d)
b = ceil(n / (d - 1)) - 1
print((b - a + 1) * (b + a) // 2)
```