

Автономные транспортные системы

Первый отборочный этап

Предметный тур. Информатика

Первая попытка. Задачи 8–11 класса

Задача II.1.1.1. Три квадрата (15 баллов)

Темы: математика, задачи для начинающих.

Условие

Фермер владеет участком земли в форме прямоугольника с длинами сторон a и b . Недавно фермер понял, что может разбить свой участок на три части так, что каждая часть будет иметь форму квадрата, и решил воспользоваться этой возможностью. Напишите программу, которая найдет площадь каждой части после разбиения.

Формат входных данных

На вход подается два натуральных числа a и b — длины сторон прямоугольника. Числа не превосходят 1000. Каждое число подается в отдельной строке. Гарантируется, что длины сторон таковы, что прямоугольник может быть разбит на три квадрата.

Формат выходных данных

Требуется вывести через пробел три натуральных числа — площади каждого из участков после разбиения. Числа могут выводиться в произвольном порядке.

Методика проверки

Программа проверяется на 12-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

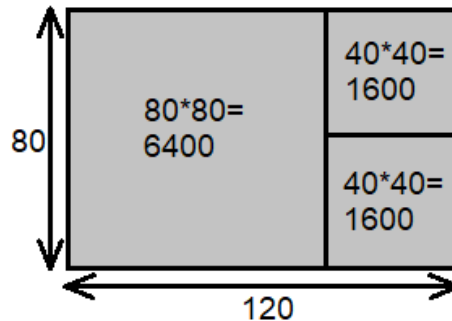
Примеры

Пример №1

Стандартный ввод
120 80
Стандартный вывод
6400 1600 1600

Пояснения к примеру

При заданных размерах прямоугольник может быть разбит на три квадрата так, как показано на рисунке ниже. Обратите внимание, что могут существовать и другие варианты разбиения.



Пример программы-решения

Ниже представлено решение на языке Python 3.

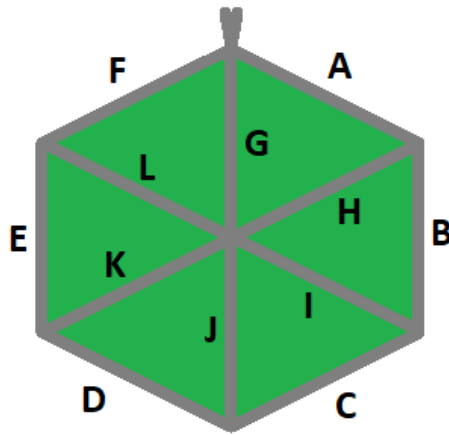
```
1 a, b = map(int, input())
2 if a > b:
3     a, b = b, a
4 s1 = a * a
5 s2 = (a * b - s1) // 2
6 print(s1, s2, s2)
```

Задача II.1.1.2. Пробежка в шестиугольном парке (18 баллов)

Темы: конструктивное построение, задачи для начинающих.

Условие

Иван Иванович совершает пробежки по парку, который имеет форму шестиугольника. В парке 12 аллей, обозначенных символами латинского алфавита от «А» до «L». Схему парка смотрите на рисунке. Длина каждой аллеи ровно 100 м. В парке есть только один вход у перекрестка аллей «А», «F», «G». Иван Иванович хочет начать и закончить пробежку у входа в парк и пробежать ровно k м. На каждом перекрестке Иван Иванович может повернуть в любую сторону, но он не хочет поворачивать назад.



Напишите программу, которая составит любой маршрут движения, удовлетворяющий указанным требованиям.

Формат входных данных

На вход в подается одно натуральное число k — желаемая длина маршрута, $300 \leq k \leq 10000$. Число k делится на 100 без остатка.

Формат выходных данных

Требуется вывести строку из $k/100$ символов, содержащую обозначения аллей в построенном маршруте.

Методика проверки

Программа проверяется на 18-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
800
Стандартный вывод
FLKEFAHG

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 a=int(input())//100
2 ans=''
3 while a>=6:
4     ans+='AHG'

```

```
5     a-=3
6     sol=['', '', '', 'AHG', 'ABIG', 'ABCJG']
7     print(ans+sol[a])
```

Задача П.1.1.3. Знакопеременная сумма (25 баллов)

Темы: префиксные суммы.

Условие

Знакопеременной суммой последовательности чисел a_1, a_2, \dots, a_k называется результат вычисления выражения $a_1 - a_2 + a_3 - a_4 + a_5 - \dots + (-1)^{k+1}a_k$. Другими словами, мы складываем все числа в последовательности, но элементы с нечетными номерами мы берем со знаком плюс, а четные со знаком минус.

Задан массив x_1, x_2, \dots, x_n из n целых чисел и m запросов. Каждый запрос содержит по два натуральных числа b и e . В ответ на каждый запрос ваша программа должна взять подмассив с номерами элементов от b до e включительно и посчитать его знакопеременную сумму $x_b - x_{b+1} + x_{b+2} - x_{b+3} + \dots + (-1)^{e-b}x_e$

Формат входных данных

На вход в первой строке подается одно натуральное число n размер массива чисел, $1 \leq n \leq 10^5$. Во второй строке через пробел записаны элемент массива целые числа x_1, x_2, \dots, x_n . Каждое из чисел не превосходит 10^6 по абсолютной величине. Далее в третьей строке записано одно натуральное число m количество запросов, $1 \leq m \leq 10^5$. В каждой из m последующих строк записано по два числа b_i и e_i таких, что $1 \leq b_i \leq e_i \leq n$. Каждая пара чисел задает границы подмассива для выполнения одного запроса.

Формат выходных данных

Требуется вывести через пробел m целых чисел s_1, \dots, s_m . Каждое из чисел должно быть равно знакопеременной сумме соответствующего подмассива.

Если вы программируете на Python, то убрать перенос строки в функции `print` можно при помощи именованного параметра `end`, например, `print(a, end='')`.

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется. В первых 10-ти тестах размер массива и количество запросов не превосходят 1000.

Примеры

Пример №1

Стандартный ввод
5
7 -4 1 3 2
4
2 2
1 3
2 4
4 5

Стандартный вывод
-4 12 -2 1

Пояснения к примеру

Для каждого из запросов ответ получается следующим образом:

$$-4 = -4$$

$$7 - (-4) + 1 = 12$$

$$-4 - 1 + 3 = -2$$

$$3 - 2 = 1$$

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n=int(input())
2 X=list(map(int,input().split()))
3 S=[0]
4 m=1
5 for x in X:
6     S.append(S[-1]+m*x)
7     m=-m
8 m=int(input())
9 ans=''
10 for i in range(3,m+3):
11     b,e=map(int,input().split())
12     s=S[e]-S[b-1]
13     if b%2==0:
14         s=-s
15     print(s,end=' ')
```

Задача II.1.1.4. Проверка корректности маршрута (25 баллов)

Темы: реализация.

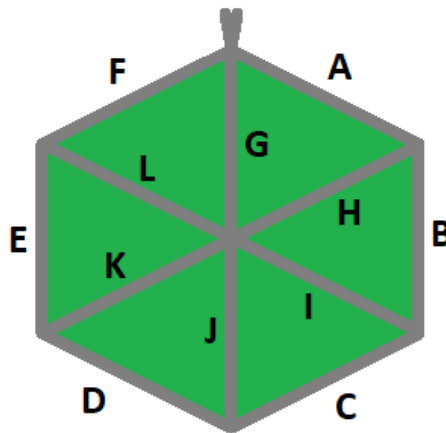
Условие

В некоторых случаях проверить корректность работы программы гораздо сложнее, чем написать ее. Сейчас у вас будет шанс в этом убедиться. От вас требуется написать программу для проверки корректности ответа второй задачи этой попытки. Напомним ее условие.

Имеется парк в виде шестиугольника с 12-ю аллеями, которые обозначены заглавными символами латиницы. В парке есть только один вход у перекрестка аллей «А», «F», «G». Схема парка приведена ниже. Требуется проверить корректность составленного маршрута движения по этому парку. Маршрут представляется как последовательность символов, представляющих аллеи в том порядке, в котором они были пройдены. Маршрут движения считается корректным если выполняются следующие требования.

- Описание маршрута содержит только символы от «А» до «L».
- Маршрут начинается и заканчивается у входа в парк.
- Запрещено разворачиваться на 180° . В частности, это означает, что начав движение с одного конца аллеи, вы обязательно дойдете до другого ее конца, причем на перекрестке вы должны будете перейти на другую аллею.

На вход вашей программе будет подано несколько описаний маршрутов. Ваша программа должна будет определить, какие из них удовлетворяют указанным требованиям.



Формат входных данных

На вход в первой строке подается одно натуральное число n — количество проверяемых маршрутов, $1 \leq n \leq 20$. Далее в n строках записаны сами маршруты. Описание каждого маршрута состоит из последовательности заглавных символов латиницы. Каждое описание не пустое, и содержит не более 100 символов.

Формат выходных данных

Программа должна вывести строку из n нулей и единиц. Единица на i -той позиции означает, что маршрут с номером i является корректным. В противном случае в этой позиции должен быть записан ноль.

Методика проверки

Программа проверяется на 5-ти тестах. Прохождение каждого теста оценивается в 5 баллов. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
6 ABCDKHA FMG ABBA ABCEF BCDEF ABCDK
Стандартный вывод
100000

Пояснения к примеру

Первый маршрут является корректным.

Второй маршрут содержит недопустимое обозначение аллеи.

В третьем маршруте происходит разворот на 180°.

Четвертый маршрут не является связным. После третьего шага он приходит к перекрестку «С», «D», «J» и с него нельзя попасть на аллею «E».

Пятый маршрут начинается не у входа.

Шестой маршрут заканчивается не у входа.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 def checkway(way):
2     c=0
3     p='Z'
4     for x in way:
5         if x<'A' or x>'L' or x==p:
6             return False
7         p=x
8         num=ord(x)-ord('A')
9         if num>5:
10            if c==6:
11                c=num-6
12            elif c==num-6:
13                c=6
14            else:
15                return False
16     else:
```

```

17         if c==num:
18             c=(c+1)%6
19         elif c==(num+1)%6:
20             c=(c+5)%6
21         else:
22             return False
23     return c==0
24
25 m=int(input())
26 for i in range(m):
27     print(int(checkway(input()))))

```

Вторая попытка. Задачи 8–11 класса

Задача П.1.2.1. Стрелки часов (12 баллов)

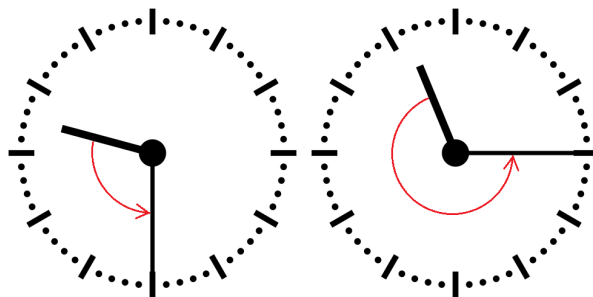
Темы: математика, задачи для начинающих.

Условие

Иван Иванович посмотрел на часы и заметил, что минутная и часовая стрелки образуют угол в α° . С этого момента прошло h ч и m мин. Напишите программу, которая вычислит угол между стрелками после истечения этого времени.

Угол измеряется от часовой до минутной стрелки в направлении против хода часов. Если стрелки совпадают, то угол равен нулю. Рассмотрим пример на рисунке. Пусть $\alpha = 105^\circ$. В частности, такой угол появляется в 9:30, так как минутная стрелка в этот момент указывает на 270° , а часовая на 165° . Через 1 ч 45 мин на часах будет 11:15. В этот момент времени минутная стрелка указывает на 0° , а часовая на $112,5^\circ$. Угол от часовой до минутной стрелки будет равен $360^\circ - 112,5^\circ = 247,5^\circ$.

Отметим, что угол $\alpha = 105^\circ$ появляется и в другие моменты времени, однако, это не повлияет на итоговый ответ.



Формат входных данных

На вход в первой строке подается одно целое неотрицательное число α исходный угол между стрелками, $0 \leq \alpha \leq 359$. Во второй строке через пробел на вход подается два числа h и m время, прошедшее с момента наблюдения в часах и минутах, $0 \leq h \leq 11$; $0 \leq m \leq 59$.

Формат выходных данных

Программа должна вывести одно вещественное число ответ к задаче. Ответ должен быть записан без погрешности.

Методика проверки

Программа проверяется на 24-х тестах. Прохождение каждого теста оценивается в 0,5 балла. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
105
1 45
Стандартный вывод
247.5

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 a = float(input())
2 h,m = map(int,input().split())
3 a = a - 5.5*(m+h*60)
4 while a<0:
5     a += 360
6 print(a)
```

Задача II.1.2.2. Номера домов (18 баллов)

Темы: реализация.

Условие

В поселке Березовом на улице Березовой построено n домов с номерами от 1 до n , причем дома с нечетными номерами расположены на одной стороне улицы, а с четными — на другой.

Рано утром дворник вышел к одному из крайних домов на этой улице, который имел номер k , и начал подметать тротуар, двигаясь от одного края улицы к другому краю. Потом он перешел на противоположную сторону улицы и начал подметать тротуар там, двигаясь назад.

Напишите программу, которая выведет номера домов, мимо которых проходил дворник, по известным числам n и k . Для лучшего понимания прочитайте пояснения к примерам.

Формат входных данных

На вход в одной строке подается два натуральных числа n и k — количество домов и номер одного из крайних домов на улице, $4 \leq n \leq 100$. Число k может принимать одно из четырех значений: $1, 2, n - 1, n$.

Формат выходных данных

Ваша программа должна вывести через пробел последовательность номеров домов, в том порядке, в котором их проходил дворник.

Если вы программируете на Python, то убрать перенос строки в функции `print` можно при помощи именованного параметра `end`, например, `print(a, end=' ')`.

Методика проверки

Программа проверяется на 36-ти тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
7 6
Стандартный вывод
6 4 2 1 3 5 7

Пример №2

Стандартный ввод
8 1
Стандартный вывод
1 3 5 7 8 6 4 2

Пояснения к примеру

Рассмотрим первый пример. На улице 7 домов, дворник вышел к дому номер 6. Это означает, что он находится на четной стороне в конце улицы, и далее он будет двигаться к ее началу, проходя мимо домов 6, 4, 2. Потом он перейдет на противоположную сторону к дому номер 1 и пойдет к концу улицы, проходя мимо домов 1, 3, 5, 7.

Во втором примере на улице 8 домов, дворник вышел к дому номер 1. Это означает, что он находится на нечетной стороне в начале улицы, и далее он будет двигаться к ее концу, проходя мимо домов 1, 3, 5, 7. Потом он перейдет на противоположную сторону к дому номер 8 и пойдет к началу улицы, проходя мимо домов 8, 6, 4, 2.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n,k=map(int,input().split())
2 if k==1:
3     print(*range(1,n+1,2),*range(n-(n%2),0,-2))
4 elif k==2:
5     print(*range(2,n+1,2),*range(n+(n%2)-1,0,-2))
6 elif k%2==1:
7     print(*range(n+(n%2)-1,0,-2),*range(2,n+1,2))
8 else:
9     print(*range(n-(n%2),0,-2),*range(1,n+1,2))
```

Задача II.1.2.3. Упорядочивание монет (25 баллов)

Темы: реализация, сортировки, теория графов.

Условие

В древнем кладе было найдено n монет различного веса. Каждая из монет была обозначена строчной буквой латиницы. Все обозначения были различными. Монеты были попарно взвешены на чашечных весах. Протокол взвешиваний состоял из $n(n-1)/2$ строк, каждая строка содержала ровно три символа. Первый и третий символ содержали обозначения монет, а во втором был записан результат сравнения: знак $<$ или знак $>$. Например, запись $d > b$ означает, что монета d тяжелее монеты b .

Взвешивания очень утомили лаборанта, и он просит вас написать программу, которая упорядочит монеты по возрастанию веса.

Формат входных данных

На вход в первой строке подается одно натуральное число n — количество монет, $4 \leq n \leq 26$. Далее в $n(n-1)/2$ строках записан протокол взвешиваний. Гарантируется, что протокол является корректным.

Формат выходных данных

Ваша программа должна вывести одну строку из n символов. Строка должна содержать обозначения монет в порядке возрастания их веса.

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
4 b<x k>b x<k b<d x>d d<k
Стандартный вывод
bdxk

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n=int(input())
2 cmp=set()
3 for i in range(n*(n-1)//2):
4     cmp.add(input())
5 lst=list({s[0] for s in cmp} | {s[2] for s in cmp})
6 for i in range(len(lst)-1):
7     for j in range(i+1,len(lst)):
8         if lst[j]+'<'+lst[i] in cmp or lst[i]+'>'+lst[j] in cmp:
9             lst[i],lst[j]=lst[j],lst[i]
10 print(''.join(lst))
```

Задача II.1.2.4. 2–3 дерево (25 баллов)

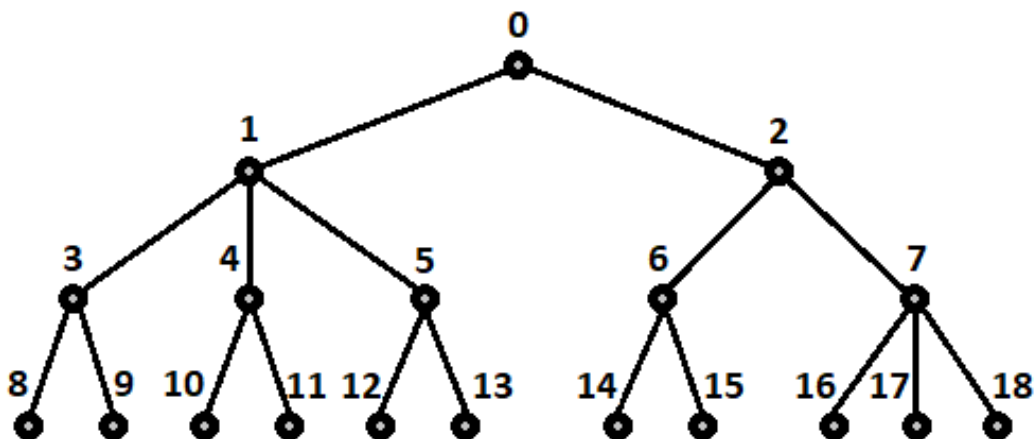
Темы: математика, графы и деревья, реализация.

Условие

Упорядоченное корневое дерево называется 2–3 деревом, если для него выполняются следующие условия:

- все узлы, кроме листьев, имеют два или три потомка;
- все листья находятся на одной высоте.

Пример 2–3 дерева приведен на рисунке ниже. Вы должны будете написать программу, которая составит произвольное 2–3 дерево с заданным количеством узлов или определит, что таких деревьев не существует.



Формат входных данных

На вход в первой строке подается одно натуральное число n — количество узлов в дереве, $2 \leq n \leq 100000$.

Формат выходных данных

Ваша программа должна вывести описание полученного дерева. Узлы дерева должны иметь номера от 0 до $n - 1$ и быть упорядоченными сверху вниз и слева направо, как на рисунке. Для каждого узла, кроме корня, требуется указать номер его непосредственного предка. Вывод состоит из последовательности чисел a_1, a_2, \dots, a_{n-1} , записанных через пробел, где a_i это непосредственный предок узла с номером i .

Можно дать и другую интерпретацию. Вывод состоит из последовательности номеров нелистовых узлов, упорядоченных по возрастанию, причем номер каждого узла повторяется столько раз, сколько у него непосредственных потомков.

Если 2–3 дерево с указанным количеством узлов построить невозможно, то требуется вывести -1 .

Методика проверки

Программа проверяется на 50-ти тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
19
Стандартный вывод
0 0 1 1 1 2 2 3 3 4 4 5 5 6 6 7 7 7

Пример №2

Стандартный ввод
5
Стандартный вывод
-1

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n=int(input())
2 k=1
3 curc=1
4 maxc=1
5 cnt=[1]
6 while maxc<n:
7     maxc+=3**k
8     curc+=2**k
9     cnt.append(2**k)
10    k+=1
11 if curc>n:
12     print('-1')
13 else:
14     for i in range(1,k):
15         t=min(3*cnt[i-1]-cnt[i],(n-curc)//(2**(k-i)-1))
16         curc+=t*(2**(k-i)-1)
17         for j in range(i,k):
18             cnt[j]+=t*(2**(j-i))
19     m=0
20     for i in range(k-1):
21         p=cnt[i]
22         s=cnt[i+1]
23         while p>0:
24             if s>2*p:
25                 print(m,m,m,end=' ')
26                 s-=3
27             else:
28                 print(m,m,end=' ')
29                 s-=2
30         p-=1
31         m+=1
```

Третья попытка. Задачи 8–11 класса

Задача П.1.3.1. Количество нечетных чисел (12 баллов)

Темы: математика, задачи для начинающих.

Условие

Задан интервал целых чисел $[b; e]$. Вы должны написать программу, чтобы определить, сколько нечетных чисел принадлежит этому интервалу.

Обратите внимание, что интервал может быть достаточно большим, и решения, перебирающие все натуральные числа, не будут проходить часть тестов.

Формат входных данных

На вход в одной строке подается два целых числа b и e — границы интервала, $-10^{18} \leq b \leq e \leq 10^{18}$.

Формат выходных данных

Программа должна вывести одно число — количество нечетных чисел в заданном интервале.

Методика проверки

Программа проверяется на 24-х тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
-3 8
Стандартный вывод
6

Пример №2

Стандартный ввод
0 0
Стандартный вывод
0

Пояснения к примеру

В первом примере указанному интервалу принадлежат шесть нечетных чисел: $-3, -1, 1, 3, 5, 7$.

Во втором примере интервал не содержит нечетных чисел.

Пример программы-решения

Ниже представлено решение на языке Python 3.

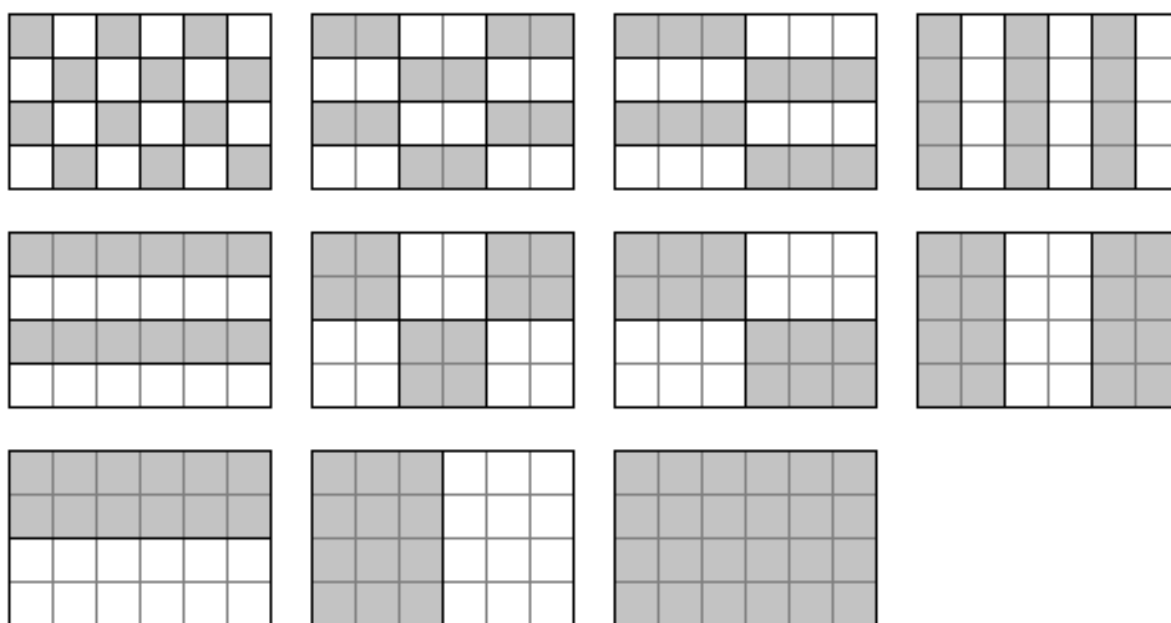
```
1 b, e = map(int, input().split())
2 print(((e+e%2)-(b-b%2))/2)
```

Задача П.1.3.2. Наибольший общий делитель прямоугольников (18 баллов)

Темы: математика, задачи для начинающих.

Условие

Будем говорить, что прямоугольник P является делителем прямоугольника Q , если прямоугольник Q можно замостить прямоугольниками P , причем все они должны иметь одинаковую ориентацию. Например, делителями прямоугольника 6×4 будут следующие прямоугольники: 1×1 , 2×1 , 3×1 , 4×1 , 6×1 , 2×2 , 3×2 , 4×2 , 6×2 , 4×3 , 6×4 . Примеры замощений можно увидеть на рисунке ниже. Обратите внимание, что прямоугольники $a \times b$ и $b \times a$ считаются одинаковыми.



Ваша задача заключается в написании программы, которая найдет наибольший общий делитель двух заданных прямоугольников. Из двух прямоугольников большим считается тот, площадь которого больше. Наибольших общих делителей может быть два. В этом случае допускается вывести любой из этих двух прямоугольников.

Формат входных данных

На вход в двух строках подаются размеры двух прямоугольников. Каждая строка содержит два натуральных числа — длину и ширину прямоугольника. Каждое из чисел не превосходит 10^{18} . Гарантируется, что введенные значения будут таковы, что площадь прямоугольника, который должен получиться в качестве ответа, не превысит 10^{18} .

Формат выходных данных

Программа должна вывести через пробел два числа — размеры искомого прямоугольника. Числа можно выводить в любом порядке.

Методика проверки

Программа проверяется на 18-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
6 4 4 3
Стандартный вывод
4 3

Пример №2

Стандартный ввод
9 10 15 3
Стандартный вывод
3 5

Пример №3

Стандартный ввод
3 7 7 3
Стандартный вывод
3 7

Пример №4

Стандартный ввод
3 7 4 8
Стандартный вывод
1 1

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 def gcd(a,b):
2     while a>0 and b>0:
3         if a>b:
4             a%=b
5         else:
```

```
6         b%=a
7     return a+b
8
9     a1,b1=map(int,input().split())
10    a2,b2=map(int,input().split())
11    a3=gcd(a1,a2)
12    b3=gcd(b1,b2)
13    a4=gcd(a1,b2)
14    b4=gcd(b1,a2)
15    if a3*b3>a4*b4:
16        print(a3,b3)
17    else:
18        print(a4,b4)
```

Задача П.1.3.3. Справедливый дележ (25 баллов)

Темы: реализация.

Условие

Два купца, живущие в разных городах, в далеком плавании купили несколько видов пряностей, и теперь хотят поделить их. Каждый из купцов будет продавать пряности только в своем городе, и цена каждой пряности в этих городах может отличаться. Купцы сочли, что будет справедливым, если они поделят пряности на две доли так, чтобы суммарная стоимость пряностей первой доли в первом городе была равна суммарной стоимости пряностей второй доли во втором городе. Существует несколько способов дележа, удовлетворяющих этому условию, но купцы хотят выбрать из них такой, при котором они получают максимум денег. Пряности являются сыпучим товаром, поэтому они могут быть поделены в любой пропорции

Рассмотрим пример. Есть три вида пряностей: перец, ваниль и корица. Стоимость всей партии перца в первом и втором городах составляет 120 и 200 условных единиц соответственно. Аналогичная стоимость партии ванили равна 180 и 140 условных единиц, а корицы — 100 и 60 условных единиц. Допустимым способом дележа будет, например, следующий: первый купец возьмет всю ваниль, второй — весь перец, а корицу они поделят поровну. Тогда стоимость доли первого купца в первом городе будет равна $180 + 100 \cdot 0,5 = 230$. Стоимость доли второго купца во втором городе составит $200 + 60 \cdot 0,5 = 230$. Стоимости долей равны, поэтому такой вариант дележа допустим. Но более выгодным будет другой вариант. Первый купец возьмет всю корицу и $3/4$ ванили, а второй купец — весь перец и $1/4$ ванили. Тогда стоимость доли в первом городе составит $100 + 180 \cdot 0,75 = 235$ и $200 + 140 \cdot 0,25 = 235$ во втором городе. Таким образом, второй вариант является более предпочтительным.

Напишите программу, которая найдет максимальную стоимость долей, при условии того, что дележ будет справедливым.

Формат входных данных

На вход в первой строке подается одно натуральное число n — количество видов пряностей, $1 \leq n \leq 100$. Во второй строке через пробел записаны n натуральных чисел a_1, a_2, \dots, a_n — цены всех видов пряностей в первом городе. Аналогично в третьей строке записаны числа b_1, b_2, \dots, b_n — цены всех видов пряностей во втором городе, $1 \leq a_i, b_i \leq 10^6$.

Формат выходных данных

Программа должна вывести одно число — максимальную стоимость долей. Это число может быть вещественным. Ответ будет считаться верным, если он отличается от ответа жюри не более чем на $0,01$.

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. В первых пяти тестах $n \leq 3$. В первых 15 тестах $n \leq 10$. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
3 120 180 100 200 140 60
Стандартный вывод
235.0

Пример №2

Стандартный ввод
1 100 200
Стандартный вывод
66.66666666666667

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n=int(input())
2 A=map(float,input().split())
3 B=map(float,input().split())
4 p=list(zip(A,B))
5 p.sort(key=lambda x:x[1]/x[0])
6 s1,s2=0,0
7 i,j=0,n-1
8 while i<=j:
9     if s1<s2:
10        s1+=p[i][0]
11        i+=1
12    else:
13        s2+=p[j][1]
14        j-=1
15 if s1<s2:
```

```

16     s1+=(s2-s1)*p[j+1][0]/(p[j+1][0]+p[j+1][1])
17 else:
18     s1--=(s1-s2)*p[i-1][0]/(p[i-1][0]+p[i-1][1])
19 print(s1)

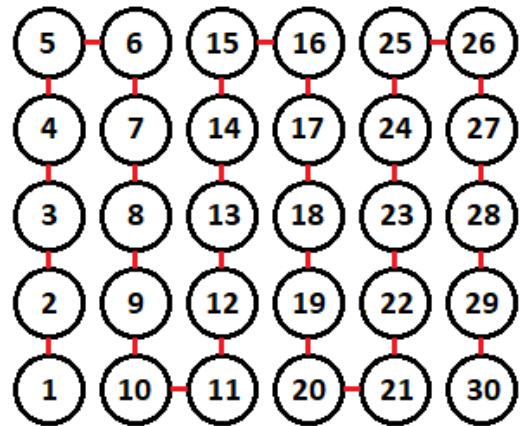
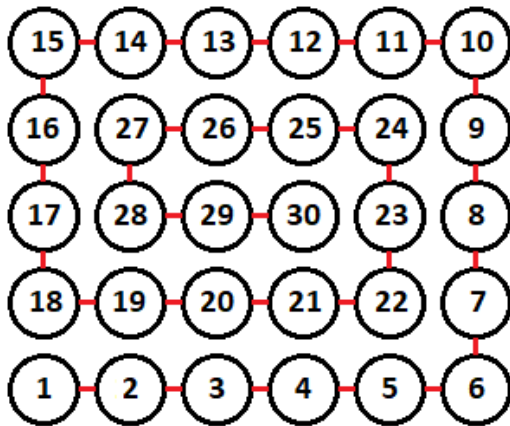
```

Задача П.1.3.4. Чай с лимоном и сахаром (25 баллов)

Темы: реализация.

Условие

На прямоугольном столе в n рядов расставлены стаканы с чаем, в каждом ряду по m стаканов. Аня ходит вокруг стола и бросает в каждый стакан по ломтику лимона. Нумерация стаканов на рисунке слева соответствует той последовательности, в которой Аня переходит от одного стакана к другому. Яна ходит вдоль одного края стола туда и обратно, и бросает в каждый стакан кусочек сахара. Нумерация стаканов на рисунке справа соответствует той последовательности, в которой Яна переходит от одного стакана к другому.



Будем считать, что ломтик лимона и кусочек сахара в один стакан девочки бросают ровно за одну секунду. Напишите программу, которая найдет количество стаканов, в которых через t с лежит и лимон и сахар. В каждом тесте ваша программа должна будет ответить на k запросов. При этом количество и расположение стаканов на столе единое для всех запросов в одном тесте.

Формат входных данных

На вход в первой строке подается два натуральных числа n , m и k — количество рядов на столе, количество кружек в каждом ряду и количество запросов, $1 \leq n, m \leq 1000$, $1 \leq k \leq 10^5$. Во второй строке через пробел записано k натуральных чисел t_1, t_2, \dots, t_k — моменты времени, для которых требуется решить задачу, $1 \leq t_i \leq nm$. Каждый момент времени может встречаться более 1 раза.

Формат выходных данных

Программа должна вывести в одной строке через пробел k чисел — ответы для каждого из заданных моментов времени.

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. В первых пятнадцати тестах $n, m \leq 10$. Тест из условия задачи при проверке не используется.

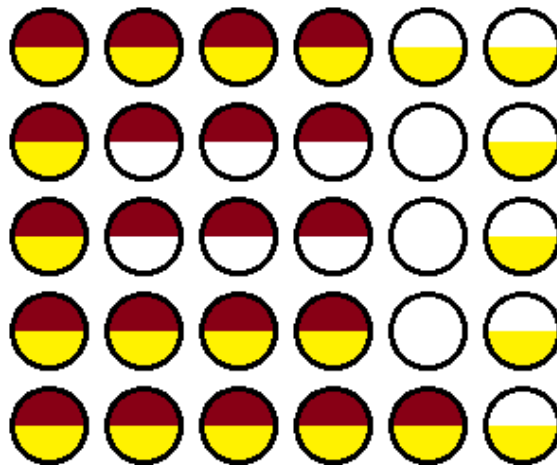
Примеры

Пример №1

Стандартный ввод
5 6 4 1 21 30 21
Стандартный вывод
1 15 30 15

Пояснения к примеру

На рисунке ниже показано решение задачи для теста из условия задачи после двадцать первой секунды. Желтым цветом помечены кружки с лимоном, коричневым — кружки с сахаром. Из рисунка видно, что в 15 чашках есть и лимон, и сахар.



Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n,m,k=map(int,input().split())
2 F=[[0 for i in range(m)] for j in range(n)]
3 A=[0]
4 i1,j1,i2,j2,s=0,0,0,0,0
5 dr=0
6 for i in range(n*m):
7     F[i1][j1]+=1
8     F[i2][j2]+=2
9     if F[i1][j1]==3:
```

```

10     s+=1
11     if (i2!=i1 or j2!=j1) and F[i2][j2]==3:
12         s+=1
13     A.append(s)
14     if dr==0:
15         if j1+1==m or (F[i1][j1+1]&1)==1:
16             dr=1
17     elif dr==1:
18         if i1+1==n or (F[i1+1][j1]&1)==1:
19             dr=2
20     elif dr==2:
21         if j1==0 or (F[i1][j1-1]&1)==1:
22             dr=3
23     else:
24         if i1==0 or (F[i1-1][j1]&1)==1:
25             dr=0
26     if dr==0:
27         j1+=1
28     elif dr==1:
29         i1+=1
30     elif dr==2:
31         j1-=1
32     else:
33         i1-=1
34     if j2%2==0:
35         if i2==n-1:
36             j2+=1
37         else:
38             i2+=1
39     else:
40         if i2==0:
41             j2+=1
42         else:
43             i2-=1
44     for i in input().split():
45         print(A[int(i)])

```

Четвертая попытка. Задачи 8–11 класса

Задача П.1.4.1. Сумма элементов списка (12 баллов)

Темы: математика, задачи для начинающих.

Условие

Алиса изучает списки в языке Python. По заданию из учебника она написала такую программу.

```

n = int(input())
x = [i%10 for i in range(n)]
print(sum(x))

```

Эта программа читает с консоли натуральное число n и делает список этой длины, состоящий из чисел от нуля до девяти, которые идут по кругу. Например, для $n = 25$ список будет иметь вид:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4]

В последней строчке на экран выводится сумма элементов этого списка. Для указанного списка, в частности, будет выведено число 100.

Боб очень недоволен учебником. Он считает, что учебник упускает главное — списки нужны для хранения информации, значимой для работы программы, а это задание можно выполнить, как минимум, без списков, а в идеале — без циклов и условий.

Напишите программу для этого задания, которую Боб сочтет удовлетворительной. Для этого она должна быстро и корректно работать для чисел до 10^{15} .

Формат входных данных

На вход подается одно натуральное число n , которое не превосходит 10^{15} .

Формат выходных данных

Программа должна вывести одно число — ответ, который напечатала бы приведенная выше программа, если бы она была способна работать со столь большими числами.

Методика проверки

Программа проверяется на 24-х тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
25
Стандартный вывод
100

Пример №2

Стандартный ввод
1000000000000000
Стандартный вывод
4500000000000000

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n = int(input())
2 return str((n//10)*45+(n%10)*(n%10-1)//2)
```

Задача П.1.4.2. Автокорреляционная функция дискретного сигнала (18 баллов)

Темы: математика, реализация.

Условие

Автокорреляционная функция часто применяется при анализе сигналов, например, энцефалограммы человека или в радиолокации. Мы будем рассматривать некоторый цифровой сигнал $a_0, a_1, a_2, \dots, a_{n-1}$, где каждое значение a_i равно 1 или -1 . Определим автокорреляционную функцию $u(t)$ по следующей формуле:

$$u(t) = \sum_{0 \leq i < n-t} a_i a_{i+t}.$$

Другими словами, если сигнал задан в виде списка из n значений, то чтобы вычислить автокорреляционную функцию в точке t , требуется взять одну копию списка без первых t элементов, другую копию списка без последних t элементов, поэлементно перемножить эти списки, и найти сумму произведений. Рассмотрим пример. Пусть сигнал содержит шесть элементов $1, 1, -1, 1, -1, 1$. Найдем $u(2)$. Исходная последовательность без первых двух элементов имеет вид $-1, 1, -1, 1$. Исходная последовательность без последних двух элементов имеет вид $1, 1, -1, 1$. Тогда $u(2) = (-1 \cdot 1) + (1 \cdot 1) + (-1 \cdot -1) + (1 \cdot 1) = 2$

По такому же принципу можно посчитать и остальные значения для t от нуля до пяти.

$$u(0) = (1 \cdot 1) + (1 \cdot 1) + (-1 \cdot -1) + (1 \cdot 1) + (-1 \cdot -1) + (1 \cdot 1) = 6$$

$$u(1) = (1 \cdot 1) + (-1 \cdot 1) + (1 \cdot -1) + (-1 \cdot 1) + (1 \cdot -1) = -3$$

$$u(3) = (1 \cdot 1) + (-1 \cdot 1) + (1 \cdot -1) = -1$$

$$u(4) = (-1 \cdot 1) + (1 \cdot 1) = 0$$

$$u(5) = 1 \cdot 1 = 1$$

Напишите программу, которая по заданному дискретному сигналу найдет значения автокорреляционной функции для всех t от 0 до $n - 1$.

Формат входных данных

На вход в первой строке подается одно натуральное число n — длина сигнала, $1 \leq n \leq 100$. Во второй строке через пробел записаны числа a_0, a_1, \dots, a_{n-1} , задающие дискретный сигнал. Каждое значение a_i равно 1 или -1 .

Формат выходных данных

Программа должна вывести через пробел n целых чисел — значения автокорреляционной функции $u(0), u(1), \dots, u(n - 1)$.

Если вы программируете на Python, то убрать перенос строки в функции `print` можно при помощи именованного параметра `end`, например, `print(a, end=' ')`.

Методика проверки

Программа проверяется на 18-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
6 1 1 -1 1 -1 1
Стандартный вывод
6 -3 2 -1 0 1

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n=int(input())
2 a=list(map(int,input().split()))
3 for k in range(n):
4     print(sum([x*y for x,y in zip(a[k:],a[:n-k])]),end=' ')
```

Задача II.1.4.3. Три фишки (25 баллов)

Темы: игры.

Условие

Алиса и Боб играют в следующую игру. Имеется игровое поле в виде последовательности клеток, расположенных друг за другом. На поле расположены три фишки, каждая фишка в своей клетке. За один ход каждый игрок должен переместить одну фишку вправо на произвольное ненулевое число клеток. При этом фишка, которой делается ход, не может встать в клетку, где расположена другая фишка или перепрыгнуть через нее. Выигрывает тот игрок, который смог сделать последний ход.

Рассмотрим пример.



Здесь возможны следующие ходы: сместить правую фишку на одну клетку; сместить среднюю фишку на одну клетку; сместить левую фишку на одну, две, три или четыре клетки.

Алиса всегда делает первый ход, а фишки расставляет Боб. Но Боб не хочет побеждать, он хочет, чтобы Алиса нашла выигрышную стратегию. Поэтому он расставляет фишки так, чтобы Алиса могла гарантированно выиграть.

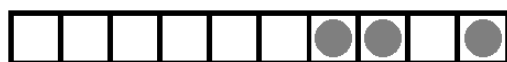
Например, в приведенной выше позиции Алиса должна сместить самую левую фишку на три клетки.



Далее игра зависит от хода Боба. Предположим, он сместит правую фишку на одну клетку. Тогда Алиса в свой ход сместит левую фишку на одну клетку.



Теперь Боб может ходить только средней фишкой. Если он сдвинет ее на одну клетку, то Алиса сдвинет левую фишку на одну клетку.



Бобу остается вновь ходить средней фишкой. Он сдвинет ее на одну клетку, Алиса сдвинет левую фишку на одну клетку и победит.



Для всех других ходов Боба у Алисы также всегда найдется ход, ведущий к победе.

Вы должны написать программу, которая по заданной позиции найдет ход, после которого Алиса сможет победить независимо от дальнейшей игры Боба. Если выигрышных ходов будет несколько, то Алиса может сделать любой из них. Напомним, что исходная позиция будет такой, что найдется как минимум один ход, гарантированно ведущий к победе.

Формат входных данных

На вход подается строка представляющая игровое поле. Пустая клетка в строке обозначена нулем, клетка с фишкой обозначена единицей. Длина строки не превосходит 1000 символов. В строке ровно три единицы.

Формат выходных данных

Программа должна вывести строку, представляющую игровое поле после хода Алисы, в том же формате, в котором она поступает на вход.

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
0100001010
Стандартный вывод
0000101010

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 x=list(reversed(input()))
2 m=[0,0,0]
3 k=0
4 for c in x:
5     if c=='0':
6         m[k]+=1
7     elif k<2:
8         k+=1
9     else:
10        break
11 if m[0]>m[2]:
12     x[m[0]]='0'
13     x[m[2]]='1'
14 else:
15     x[m[0]+m[1]+m[2]+2]='0'
16     x[2*m[0]+m[1]+2]='1'
17 print(''.join(reversed(x)))
```

Задача II.1.4.4. Выбор купюр (25 баллов)

Темы: реализация.

Условие

В денежной системе Бурляндии выпускаются банкноты всех номиналов от a до $2a$ включительно. У Алисы в бумажнике есть ровно одна банкнота каждого номинала. Алиса хочет сделать покупку ценой b и расплатиться без сдачи. Кроме того, Алиса хочет, чтобы количество потраченных банкнот было как можно меньшим. Напишите программу, которая поможет Алисе выбрать банкноты так, чтобы сумма их номиналов была равна b , а их количество было наименьшим среди возможных. Если указанным условиям удовлетворяет несколько наборов банкнот, то ваша программа может вывести любой из них.

Формат входных данных

На вход в одной строке подается два натуральных числа a и b — минимальный из номиналов купюр и требуемая сумма, $1 \leq a \leq 100000$. Гарантируется, что для заданной суммы b существует способ получить ее из имеющихся купюр.

Формат выходных данных

Программа должна вывести в одной строке через пробел номиналы всех банкнот, которые потребуются для оплаты. *Все номиналы должны быть упорядочены по возрастанию.*

Методика проверки

Программа проверяется на 25-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
10 99
Стандартный вывод
10 15 17 18 19 20

Пояснения к примеру

Сумма чисел, указанных в ответе, равна 99, и все числа лежат в диапазоне от 10 до 20 включительно. При этом сумма номиналов пяти самых ценных банкнот меньше чем 99, поэтому оплатить указанную сумму пятью или меньшим числом банкнот невозможно. Однако другие варианты получения требуемой суммы шестью банкнотами возможны, например, 13 14 15 18 19 20. Такой ответ тоже будет засчитан.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 a,b=map(int,input().split())
2 up,dn,k=0,0,0
3 while up<b:
4     up+=2*a-k
5     dn+=a+k
6     k+=1
7
8 if a>k-1:
9     t=(b-dn)//(a-k+1)
10    r=(b-dn)%(a-k+1)
11 else:
12    t,r=0,0
13 if t==k:
14    print(*range(2*a-t+1,2*a+1))
15 else:
16    print(*range(a,a+k-t-1),a+k-1-t+r,*range(2*a-t+1,2*a+1))
```

Автономные транспортные системы

Первый отборочный этап

Предметный тур. Физика

Первая попытка. Задачи 8–9 класса

Задача П.2.1.1. Увлажнитель воздуха (15 баллов)

Темы: тепловой баланс, испарение.

Условие

Паровой увлажнитель воздуха нагревает воду и доводит ее до кипения. С помощью электрического нагревателя увлажнитель испаряет q мл воды за 1 ч, q мл/ч. Найти мощность нагревателя $P_{\text{наг}}$. Начальная температура воды t_0 °С. Удельная теплоемкость воды $C = 4,2$ кДж/(кг · °С), удельная теплота парообразования $L = 2,3$ МДж/кг. Плотность воды $\rho = 1000$ кг/м³. Ответ дать в Вт с точностью 1 Вт.

Решение

За время τ увлажнитель испаряет воду массой $\rho q \tau$.

Уравнение теплового баланса за это время:

$$P_{\text{наг}} \tau = q \rho \tau [C(t_{\text{кип}} - t_0) + L], \text{ где } t_{\text{кип}} = 100^\circ\text{C}.$$

Отсюда:

$$P_{\text{наг}} = q \rho [C(t_{\text{кип}} - t_0) + L].$$

Погрешность 2 Вт.

Диапазоны

Величина	min	max	Шаг
t_0	16	24	1
q	200	300	5

Ответ: $q \rho [C(t_{\text{кип}} - t_0) + L]$.

Задача П.2.1.2. Электропоезд «Ласточка» начинает двигаться (20 баллов)

Темы: КПД.

Условие

Электропоезд «Ласточка» состоит из пяти вагонов общей массой m т. Два моторных вагона (с электродвигателями) создают общую силу тяги $F = 280$ кН, благодаря которой поезд начинает двигаться с места с ускорением a м/с². Найти КПД электропоезда. Ответ дать в % с точностью 1 %.

Решение

Спустя время t после начала движения кинетическая энергия поезда равна $\frac{m(at)^2}{2}$, а работа силы тяги равна $\frac{Fat^2}{2}$. Тогда КПД электропоезда равен отношению полезной работы к затраченной:

$$\eta = \frac{ma}{F} \times 100\%.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
a	0,60	0,70	0,01
m	240	260	1

Ответ: $\eta = \frac{ma}{F} \times 100\%$.

Задача II.2.1.3. Электропитание оптических модулей (25 баллов)

Темы: закон Ома.

Условие



Для глубоководного нейтринного телескопа на Байкале используют кластеры, состоящие из восьми гирлянд. Каждая гирлянда — это уходящий под воду вертикальный тонкий трос с навешенными оптическими модулями (шарами) для регистрации световых вспышек. Нижний конец троса укреплен на дне с помощью якоря, к верхнему концу троса прикреплен подводный буй, который поддерживает трос в вертикальном положении. Электропитание каждого оптического модуля осуществляется источником питания с напряжением $U = 13$ В. Оптическому модулю для работы требуется напряжение U_0 В, и он подключается к источнику питания медным проводом общей длиной l м, по которому протекает постоянный ток $I = 250$ мА. Найти площадь сечения медного провода S . Удельное сопротивление меди $\rho = 0,017$ Ом·мм²/м. Ответ дать в мм² с точностью 0,01 мм².

Решение

Напряжение на медном проводе равно $U - U_0$. По закону Ома сопротивление медного провода:

$$\frac{U - U_0}{I} = \frac{\rho l}{S}.$$

Отсюда площадь сечения медного провода равна:

$$S = \frac{\rho l I}{U - U_0}.$$

Погрешность 0,02 мм².

Диапазоны

Величина	min	max	Шаг
U_0	11,5	12,2	0,1
l	170	190	1

Ответ: $\frac{\rho l I}{U - U_0}$.

Задача II.2.1.4. Автомобиль с неисправностью (25 баллов)

Темы: мощность.

Условие

Незнайка с помощью волшебной палочки сделал обычный автомобиль, двигатель которого работал на спирту. КПД его двигателя оказался равным $\eta_1\%$. После длительного путешествия КПД уменьшился до значения $\eta_2\%$. Оказалось, что часть спирта стала вытекать через трещину в шланге. Найти долю δ вытекающего спирта. Ответ дать в % с точностью 0,1 %.

Решение

КПД исправного двигателя равен отношению его работы A к тепловой энергии Q сгораемого спирта: $\eta_1 = \frac{A}{Q}$. В двигателе с утечкой спирта работа уменьшается в $(1 - \delta/100\%)$ раз. Поэтому $\eta_2 = \frac{(1 - \delta/100\%)A}{Q}$. Отсюда находим долю δ вытекающего спирта:

$$\delta = \left(1 - \frac{\eta_2}{\eta_1} \times 100\%\right).$$

Погрешность 0,2%.

Диапазоны

Величина	min	max	Шаг
η_1	24,0	26,2	0,1
η_2	20,0	22,5	0,1

Ответ: $\delta = \left(1 - \frac{\eta_2}{\eta_1} \times 100\%\right)$.

Задача II.2.1.5. (15 баллов)

Темы: простейшая динамика, силы.

Условие

К деревянному бруску массой 400 г, лежащему на горизонтальной деревянной доске, прикрепили пружину жесткостью 50 Н/м. Ученик потянул за пружину в горизонтальном направлении и брусок начал равномерно двигаться. Определить удлинение пружины, если коэффициент трения дерева по дереву равен 0,2.

Ответ выразить в мм и округлить до целых. Ускорение свободного падения считать равным 10 м/с^2

Решение

Уравнения движения (2-й закон Ньютона) в проекциях на горизонтальную и вертикальную оси выглядят так: $kx = F_{\text{тр}}$. $mg = N$.

Кроме того, $F_{\text{тр}} = \mu N$. В итоге $x = \frac{\mu mg}{k} = 16 \text{ мм}$.

Ответ: 16.

Вторая попытка. Задачи 8–9 класса

Задача II.2.3.1. Электропоезд «Ласточка» (15 баллов)

Темы: мощность электрического тока.

Условие

Электропоезд «Ласточка» состоит из пяти вагонов. В поезде работают n электродвигателей общей мощностью $P = 2932$ кВт. Питание электропоезда происходит через контактную сеть напряжением U кВ. Найти силу тока, поступающего к каждому параллельно подключенному двигателю. Ответ дать в А с точностью 1 А.

Решение

Из выражения для мощности электрического тока получаем

$$I = \frac{P}{nU}.$$

Погрешность 1 А.

Диапазоны

Величина	min	max	Шаг
n	4	8	1
U	2,8	3,2	0,1

Ответ: $\frac{P}{nU}$.

Задача II.2.3.2. КПД увлажнителя воздуха (20 баллов)

Темы: тепловой баланс, КПД.

Условие

Паровой увлажнитель воздуха номинальной мощностью 260 Вт работает в комнате. С помощью электрического нагревателя увлажнитель нагревает и испаряет 300 мл воды за 1 ч, $q = 300$ мл/ч. Найти КПД увлажнителя.

Удельная теплоемкость воды $C = 4,2$ кДж/(кг · °С), удельная теплота парообразования $L = 2,3$ МДж/кг. Температура в комнате $t_0 = 20$ °С. Плотность воды $\rho = 1000$ кг/м³. Ответ дать в процентах с точностью 1%.

Решение

За время τ увлажнитель испаряет воду массой $\rho q \tau$.

Уравнение теплового баланса за это время:

$$P_{\text{наг}} \tau = q \rho \tau [C(t_{\text{кип}} - t_0) + L], \text{ где } t_{\text{кип}} = 100^\circ\text{C}.$$

Отсюда находим мощность нагревателя:

$$P_{\text{наг}} = q \rho [C(t_{\text{кип}} - t_0) + L].$$

Тогда

$$\text{КПД} = \frac{P_{\text{наг}}}{P} \times 100\% = \frac{q\rho [C(t_{\text{кип}} - t_0) + L]}{P} \times 100\% = 84\%.$$

Погрешность 1%.

Ответ: 84 ± 1 .

Задача П.2.3.3. Сила натяжения троса (25 баллов)

Темы: сила Архимеда, второй закон Ньютона.

Условие



Для глубоководного нейтринного телескопа на Байкале используют кластеры, состоящие из восьми гирлянд. Каждая гирлянда — это уходящий под воду вертикальный тонкий трос с навешенными оптическими модулями (шарами массой $m_1 = 15$ кг и объемом $V_1 = 0,065$ м³). Нижний конец троса укреплен на дне с помощью якоря, к верхнему концу троса прикреплен подводный буй, который поддерживает трос в вертикальном положении. Буй вместе с участком троса до верхнего шара имеет массу M_1 кг, а участок троса от верхнего шара до нижнего шара имеет массу M_2 кг. Всего на тросе закреплено $n = 36$ шаров на одинаковых расстояниях. К нижнему шару прикреплен участок троса с якорем. Найти силу натяжения троса в точке чуть ниже прикрепления шара с номером $n_1 = 10$. Шары нумеруются вдоль троса сверху вниз. Объем буя $V = 1,8$ м³.

Плотность воды $\rho = 1000$ кг/м³. Ускорение свободного падения $g = 9,8$ м/с². Ответ дать в Н с точностью 1 Н.

Решение

Выше этой точки находится буй со своим тросом, n_1 шаров и $n_1 - 1$ одинаковых кусков троса между соседними шарами. Из 2-го закона Ньютона для этой совокуп-

ности находим силу натяжения троса:

$$T = \rho g (V + n_1 V_1) - \left(M_1 + M_2 \frac{n_1 - 1}{n - 1} + n_1 m_1 \right) g.$$

Погрешность 10 Н.

Диапазоны

Величина	min	max	Шаг
M_1	1450	1550	5
M_2	1050	1100	5

Ответ: $\rho g (V + n_1 V_1) - \left(M_1 + M_2 \frac{n_1 - 1}{n - 1} + n_1 m_1 \right) g.$

Задача II.2.3.4. Электромобиль попал в аварию (25 баллов)

Темы: мощность, КПД.

Условие

Незнайка с помощью волшебной палочки сделал современный электромобиль, в котором используется электродвигатель постоянного тока, работающий от аккумулятора. КПД электродвигателя η . Однажды Незнайка с друзьями отправился на электромобиле в путешествие. Рабочий ток в обмотке электродвигателя был равен $I_1 = 0,45$ А. Неожиданно электромобиль наехал на кочку, и электродвигатель перестал вращаться. Ток в обмотке электродвигателя увеличился до значения $I_2 = 2,15$ А. Найти η . Ответ дать в % с точностью 1%.

Решение

При работе электромотора его мощность расходуется на полезную мощность P и мощность тепловых потерь:

$$UI = P + I^2 R,$$

где U — напряжение аккумулятора, R — сопротивление обмотки. Отсюда получим два уравнения для двух токов

$$UI_1 = P + I_1^2 R,$$

$$UI_2 = I_2^2 R.$$

Из этой системы находим КПД электродвигателя

$$\eta = \frac{P}{UI_1} = \left(1 - \frac{I_1}{I_2} \right) \times 100\% = 79\%.$$

Погрешность 1%.

Ответ: 79 ± 1 .

Задача II.2.3.5. (15 баллов)

Темы: кинематика.

Условие

Оцените продолжительность полета к звезде Тау Кита, находящейся на расстоянии 12 световых лет от Солнечной системы, на космическом корабле, летящем со скоростью 30 км/с. 1 св. год = $9,5 \cdot 10^{12}$ км. Ответ необходимо дать в тыс. лет.

Решение

За время полета корабля t_k он должен преодолеть расстояние $S = c \cdot t_{св}$, то есть:

$$c \cdot t_{св} = V_k \cdot t_k \Rightarrow t_k = \frac{c t_{св}}{V_k} = \frac{3 \cdot 10^5 \text{ км/с} \cdot 12 \text{ лет}}{30 \text{ км/с}} = 12 \cdot 10^4 \text{ лет} = 120 \text{ тыс. лет.}$$

Ответ: 120 ± 2000 .

Третья попытка. Задачи 8–9 класса

Задача II.2.5.1. Буй гирлянды (20 баллов)

Темы: сила Архимеда.

Условие



Для глубоководного нейтринного телескопа на Байкале используют кластеры, состоящие из восьми гирлянд. Каждая гирлянда – это уходящий под воду вертикальный тонкий трос с навешенными оптическими модулями (шарами массой $m_1 = 15$ кг и объемом $V_1 = 0,065$ м³). Нижний конец троса укреплен на дне с помощью якоря, к верхнему концу троса прикреплен подводный буй, который поддерживает трос в вертикальном положении. Буй вместе с участком троса до верхнего шара имеет массу M_1 кг, а участок троса от верхнего шара до нижнего шара имеет массу M_2 кг. Всего на тросе закреплено $n = 36$ шаров на одинаковых расстояниях. К нижнему шару прикреплен участок троса с якорем. В этой точке прикрепления сила натяжения троса равна $T = 10000$ Н. Найти объем буя V . Плотность воды $\rho = 1000$ кг/м³. Ускорение свободного падения $g = 9,8$ м/с². Ответ дать в м³ с точностью 0,01 м³.

Решение

На часть гирлянды, расположенной выше точки прикрепления участка троса с якорем, действуют сила тяжести, сила Архимеда и сила натяжения троса:

$$(M_1 + M_2 + nm_1)g + T = \rho g(V + nV_1).$$

Отсюда находим объем буя

$$V = \frac{M_1 + M_2 + nm_1}{\rho} + \frac{T}{\rho g} - nV_1.$$

Погрешность 0,02 м³.

Диапазоны

Величина	min	max	Шаг
M_1	1450	1550	5
M_2	1050	1100	5

Ответ: $\frac{M_1 + M_2 + nm_1}{\rho} + \frac{T}{\rho g} - nV_1.$

Задача II.2.5.2. Сила трения (25 баллов)

Темы: второй закон Ньютона.

Условие

Электропоезд «Ласточка» состоит из пяти вагонов общей массой m т. Каждый из двух моторных вагонов создает силу тяги $F_1 = 140$ кН, благодаря которой поезд начинает двигаться с места с ускорением a м/с². Найти силу трения, действующую на поезд. Силой сопротивления воздуха пренебречь. Ответ дать в кН с точностью 1 кН.

Решение

2-ой закон Ньютона:

$$ma = 2F_1 - F_{\text{тр}}.$$

Отсюда

$$F_{\text{тр}} = 2F_1 - ma.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
m	240	260	1
a	0,60	0,70	0,01

Ответ: $2F_1 - ma$.

Задача II.2.5.3. Повышение температуры воздуха (0 баллов)

Темы: тепловой баланс, испарение и нагревание.

Условие



Паровой увлажнитель воздуха номинальной мощностью 260 Вт работает в закрытой комнате объемом $V \text{ м}^3$. С помощью электрического нагревателя увлажнитель нагревает и испаряет 300 мл воды за 1 ч: $q = 300 \text{ мл/ч}$. Начальная температура воздуха в комнате $t_0 = 20 \text{ }^\circ\text{C}$. Удельная теплоемкость воды $C = 4,2 \text{ кДж}/(\text{кг} \cdot \text{ }^\circ\text{C})$, удельная теплота парообразования $L = 2,3 \text{ МДж}/\text{кг}$. Плотность воды $\rho = 1000 \text{ кг}/\text{м}^3$.

Найти температуру воздуха в комнате t через время T мин после начала работы увлажнителя. Плотность воздуха $\rho = 1,29 \text{ кг}/\text{м}^3$, а его удельная теплоемкость равна $C_{\text{воз}} = 4,2 \text{ кДж}/(\text{кг} \cdot \text{ }^\circ\text{C})$. Теплоемкостью увлажнителя, водяного пара и стен комнаты пренебречь. Ответ дать в $^\circ\text{C}$ с точностью 0,1 $^\circ\text{C}$.

Решение

За время T увлажнитель испаряет воду массой ρqT .

Уравнение теплового баланса за это время:

$$P_{\text{наг}}T = q\rho T [C(t_{\text{кип}} - t_0) + L],$$

где $t_{\text{кип}} = 100^\circ\text{C}$.

Отсюда находим мощность, затраченную на испарение воды

$$P_{\text{наг}} = q\rho [C(t_{\text{кип}} - t_0) + L].$$

Тогда получаем температуру воздуха в комнате через время T мин после начала работы увлажнителя:

$$t = t_0 + \frac{(P - P_{\text{наг}})T}{C_{\text{воз}}\rho_{\text{воз}}V}.$$

Погрешность $0,2 \text{ } ^\circ\text{C}$.

Диапазоны

Величина	min	max	Шаг
T	200	300	10
V	80	120	5

Ответ: $t_0 + \frac{(P - P_{\text{наг}})T}{C_{\text{воз}}\rho_{\text{воз}}V}.$

Задача II.2.5.4. Электромобиль (35 баллов)

Темы: мощность.

Условие

Незнайка с помощью волшебной палочки сделал современный электромобиль, в котором используется электродвигатель постоянного тока мощностью P Вт, работающий от аккумулятора с напряжением $U = 12$ В. Рабочий ток в обмотке электродвигателя равен I А. Найти сопротивление R обмотки электродвигателя. Ответ дать в Ом с точностью $0,1$ Ом.

Решение

При работе электромотора его мощность расходуется на полезную мощность и мощность тепловых потерь:

$$UI = P + I^2R.$$

Отсюда находим сопротивление R обмотки электродвигателя

$$R = \frac{UI - P}{I^2}.$$

Погрешность $0,2$ Ом.

Величина	min	max	Шаг
P	3,0	4,2	0,1
I	0,44	0,55	0,1

Ответ: $\frac{UI - P}{I^2}.$

Задача II.2.5.5. (20 баллов)

Темы: кинематика.

Условие

Астроном наблюдает в телескоп ближайшую к нам звезду звездную систему Альфа Центавра, которая находится на расстоянии $4,05 \cdot 10^{13}$ км от Солнечной системы. Какой он видит эту звезду? Такой, какая она есть сейчас, или такой, какая она была в прошлом? В ответе указать 0 лет, если астроном видит звезду такая она есть сейчас, или то количество лет (приблизительно) какой она была в прошлом. При этом помним о том, что свет распространяется со скоростью 300 тыс. км/с. Ответ дать с точностью до десятых года.

Решение

От звезды до астронома свет идет время

$$t = \frac{S}{c} = \frac{4,05 \cdot 10^{13} \text{ км}}{3 \cdot 10^5 \text{ км/с}} = 1,35 \cdot 10^8 \text{ с} \approx 4,28 \text{ лет},$$

то есть астроном видит звезду такой, какой она была примерно 4,28 года назад.

Ответ: $4,3 \pm 0,1$.

Четвертая попытка. Задачи 8–9 класса

Задача II.2.7.1. Автомобиль (15 баллов)

Темы: КПД.

Условие

Незнайка с помощью волшебной палочки сделал обычный автомобиль, двигатель которого работал на спирту. В спиртовом двигателе внутреннего сгорания расход топлива равен $q = 1,05$ мкг/с. Удельная теплота сгорания спирта $r = 26$ МДж/кг. За время t мин все отданное двигателем тепло Q Дж получает система охлаждения двигателя. Найти КПД спиртового двигателя. Ответ дать в % с точностью 1%.

Решение

От нагревателя получается тепло qrt и отдается холодильнику тепло Q . Тогда:

$$\text{КПД} = \left(1 - \frac{Q}{qrt}\right) \times 100\%.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
t	10,5	11,5	0,1
Q	11,0	11,7	0,1

Ответ: $\left(1 - \frac{Q}{qrt}\right) \times 100\%$.

Задача П.2.7.2. Повышение влажности воздуха (20 баллов)

Темы: влажность воздуха.

Условие

Паровой увлажнитель воздуха номинальной мощностью 260 Вт работает в закрытой комнате объемом $V = 100 \text{ м}^3$. С помощью электрического нагревателя увлажнитель испаряет q мл воды за 1 ч, q мл/ч. Постоянная температура в комнате $t_0 = 20 \text{ }^\circ\text{C}$. Плотность насыщенного водяного пара при этой температуре $\rho_{\text{нас}} = 17,3 \text{ г/м}^3$. Начальная влажность воздуха $\varphi_0\%$. Плотность воды $\rho = 1000 \text{ кг/м}^3$. Найти влажность воздуха $\varphi\%$ через время $T = 50$ мин после начала работы увлажнителя. Ответ дать в % с точностью 1%.

Решение

Начальная плотность водяного пара в воздухе равна $\varphi_0\rho_{\text{нас}}/100\%$. За время T увлажнитель испаряет воду массой ρqT . Тогда плотность водяного пара увеличится на величину $\rho qT/V$ и влажность воздуха станет равной:

$$\varphi = \frac{\varphi_0\rho_{\text{нас}}/100\% + \rho qT/V}{\rho_{\text{нас}}} \times 100\%.$$

Диапазоны

Величина	min	max	Шаг
φ_0	20	35	1
q	200	300	5

Ответ: $\frac{\varphi_0\rho_{\text{нас}}/100\% + \rho qT/V}{\rho_{\text{нас}}} \times 100\%$.

Задача П.2.7.3. КПД оптического модуля (25 баллов)

Темы: закон Джоуля – Ленца, КПД.

Условие



Для глубоководного нейтринного телескопа на Байкале используют кластеры, состоящие из восьми гирлянд. Каждая гирлянда — это уходящий под воду вертикальный тонкий трос с навешенными оптическими модулями (шарами) для регистрации световых вспышек. Нижний конец троса укреплен на дне с помощью якоря, к верхнему концу троса прикреплен подводный буй, который поддерживает трос в вертикальном положении. Электропитание каждого оптического модуля сопротивлением R Ом осуществляется низковольтным источником питания с напряжением U . Оптическому модулю для работы требуется напряжение $U_0 < U$ и он подключается к источнику питания медным проводом общей длиной l м, по которому протекает постоянный ток I . Площадь сечения медного провода $S = 1,5 \text{ мм}^2$. Удельное сопротивление меди $\rho = 0,017 \text{ Ом} \cdot \text{мм}^2/\text{м}$. Найти КПД оптического модуля. Ответ дать в % с точностью 1%.

Решение

Сопротивление медного провода равно: $\frac{\rho l}{S}$. КПД равно отношению полезной работы к затраченной. С использованием закона Джоуля – Ленца получим:

$$\text{КПД} = \frac{I^2 R}{I^2 (R + \rho l / S)} \times 100\% = \frac{100\%}{1 + \rho l / (SR)}$$

Погрешность 2%.

Диапазоны

Величина	min	max	Шаг
R	14,0	16,0	0,1
l	170	190	1

Ответ: $\frac{100\%}{1 + \rho l / (SR)}$.

Задача II.2.7.4. Постоянная скорость поезда (25 баллов)

Темы: второй закон Ньютона, равномерное движение.

Условие

Электропоезд «Ласточка» состоит из пяти вагонов общей массой $m = 260$ т. Каждый из двух моторных вагонов создает силу тяги $F_1 = 120$ кН, благодаря которой поезд движется с постоянной скоростью V . Сила трения, действующая на поезд, равна $F_{\text{тр}} = 80$ кН. Сила сопротивления воздуха пропорциональна скорости поезда $F_c = -kV$, где коэффициент сопротивления равен $k = 4,0$ кН/(м/с). Найти скорость поезда. Ответ дать в м/с с точностью 1 м/с.

Решение

При движении с постоянной скоростью сумма проекций горизонтальных сил равна нулю:

$$2F_1 - F_{\text{тр}} - F_c = 0.$$

Отсюда находим постоянную скорость:

$$V = \frac{2F_1 - F_{\text{тр}}}{k} = 40 \text{ м/с}.$$

Погрешность 1 м/с.

Ответ: 40 ± 1 .

Задача II.2.7.5. (15 баллов)

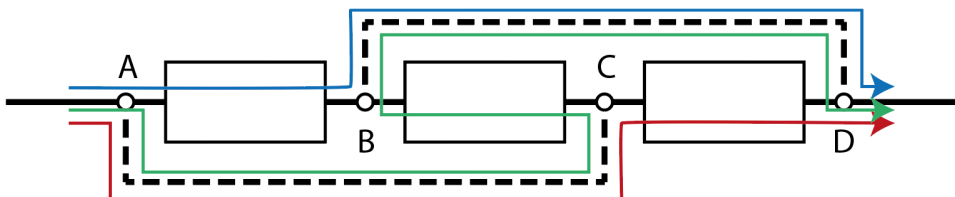
Темы: электродинамика.

Условие

Три прибора с одинаковым сопротивлением распаяны на плате с четырьмя контактами: A , B , C , D . Первый прибор включен между контактами A и B , второй — между контактами B и C , третий — между контактами C и D . При этом сопротивление между контактами A и D оказывается равно R_0 . Во сколько раз уменьшится это сопротивление, если впаять два соединительных провода с пренебрежимо малым сопротивлением: один, соединяющий контакты A и C , и второй, соединяющий контакты B и D ? Ввести ответ одним числом.

Решение

Изобразив схему изначального соединения приборов, несложно увидеть, что она представляет собой простое последовательное соединение. Добавление проводов AC и BD (пунктирные на рисунке) создаёт схему, эквивалентную параллельному соединению, поскольку ток может пройти между контактами A и D через любой из включённых приборов независимо (соответствующие «траектории» изображены на схеме разными цветами), а потенциалы точек A , C и B , D соответственно, оказываются равны. Используя формулы для эффективного сопротивления последовательно и параллельно подключённых резисторов, легко показать, что сопротивление цепи уменьшится в 9 раз.



Ответ: 9.

Автономные транспортные системы

Первый отборочный этап

Предметный тур. Физика

Первая попытка. Задачи 10–11 класса

Задача II.2.2.1. Ведро воды (15 баллов)

Темы: масса вещества.

Условие

Десятиклассник наливает воду в пластмассовое ведро. Ведро имеет форму усеченного конуса высотой h см, нижним диаметром d_1 см и верхним диаметром $d_2 = 25$ см. Какая масса воды заполнит ведро целиком? Плотность воды $\rho = 1000$ кг/м³. Ответ дать в кг с точностью 0,1 кг. Объем прямого конуса равен третьей части от произведения площади основания на высоту.

Решение

Объем ведра равен:

$$V = \frac{\pi h(d_1^2 + d_1 d_2 + d_2^2)}{12}.$$

Масса воды:

$$m = \frac{\pi h \rho (d_1^2 + d_1 d_2 + d_2^2)}{12}.$$

Погрешность 0,1 кг.

Диапазоны

Величина	min	max	Шаг
d_1	14	24	1
h	22	32	1

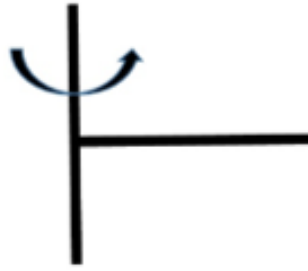
Ответ: $\frac{\pi h \rho (d_1^2 + d_1 d_2 + d_2^2)}{12}$.

Задача II.2.2.2. Два вращающихся прута (20 баллов)

Темы: равномерное вращение, центробежная сила.

Условие

Два тонких стальных прута соединили под прямым углом так, что к середине первого прута приварен конец второго прута. Первый прут расположили вдоль вертикальной оси, а второй прут вращается в горизонтальной плоскости с постоянной угловой скоростью $\omega = 3,1$ рад/с. Найти горизонтальную составляющую силы, действующей на первый прут со стороны второго прута. Масса второго прута m кг, а его длина L м. Ответ дать в Н с точностью 1 Н.



Решение

На второй прут действует горизонтальная центробежная сила. Ее можно найти, если мысленно разрезать прут на небольшие кусочки длиной Δl . На такой кусочек массой $\frac{m\Delta l}{L}$, находящийся на расстоянии l от оси вращения, действует небольшая центробежная сила $\Delta F = \frac{m\Delta l}{L}\omega^2 l$. Сложив все эти силы, получим полную центробежную силу $\left(\sum_0^L l\Delta l = \frac{L^2}{2}\right)$:

$$F = m\omega^2 L/2.$$

Эта же сила действует на первый прут

Погрешность 1 Н.

Диапазоны

Величина	min	max	Шаг
m	3,0	7,0	0,5
L	1,3	1,7	0,1

Ответ: $m\omega^2 L/2$.

Задача П.2.2.3. Полет паука (20 баллов)

Темы: второй закон Ньютона, ускорение.

Условие

Маленький паук размером 1 мм и массой $m = 5,0$ мг может подняться в воздух за счет электрического заряда на нитях своей паутины $q = 1,0$ нКл. Между землей и облаками постоянно существует электрическое поле, напряженность которого у поверхности земли на расстояниях, сравнимых с размером паука, может достигать больших значений. Известно, что после вертикального взлета паук летел равноускорено, и за время t с поднялся на высоту h см. Найти значение напряженности электрического поля E в этом случае. Ускорение свободного падения $g = 9,8$ м/с². Спротивлением воздуха пренебречь. Ответ дать в кВ/м с точностью 1 кВ/м.

Решение

Вертикальное ускорение паука $a = 2h/t^2$. Уравнение движения паука:

$$ma = qE - mg.$$

Отсюда находим значение напряженности электрического поля:

$$E = m(g + 2h/t^2)/q.$$

Погрешность 1 кВ/м.

Диапазоны

Величина	min	max	Шаг
h	0,15	0,25	0,01
t	0,045	0,055	0,001

Ответ: $m(g + 2h/t^2)/q$.

Задача II.2.2.4. Электромобиль (25 баллов)

Темы: мощность.

Условие

Винтик и Шпунтик смастерили для коротышек современный электромобиль. Они использовали электродвигатель постоянного тока, работающий от аккумулятора. КПД электродвигателя $\eta = 0,69$. Однажды Незнайка забрался в электромобиль, нажал на кнопку, и машина поехала. Рабочий ток в обмотке электродвигателя был равен I_1 . Затем электромобиль наехал на собачью будку и электродвигатель перестал вращаться. Ток в обмотке электродвигателя увеличился до значения I_2 , а напряжение на аккумуляторе осталось неизменным. Найти отношение токов I_2/I_1 . Ответ дать с точностью 0,1.

Решение

При работе электромотора его мощность расходуется на полезную мощность P и мощность тепловых потерь:

$$UI = P + I^2R,$$

где U — напряжение аккумулятора, R — сопротивление обмотки. Отсюда получим два уравнения для двух токов:

$$UI_1 = P + I_1^2R,$$

$$UI_2 = I_2^2R.$$

Из этой системы находим КПД электродвигателя:

$$\eta = \frac{P}{UI_1} = 1 - \frac{I_1}{I_2}.$$

Отношение токов равно $I_2/I_1 = 1/(1 - \eta) = 3,2$.

Погрешность 0,1.

Ответ: $3,2 \pm 0,1$.

Задача II.2.2.5. (15 баллов)

Темы: энергия, закон сохранения энергии.

Условие

С какой высоты должна была бы падать на землю градина при начальной температуре -10°C , чтобы после падения на земле оставалась лужа при 0°C , если известно, что в процессе падения $\alpha = 15\%$ механической энергии градины уходит на нагревание окружающего ее воздуха? Удельная теплоемкость льда равна $c = 2100 \text{ Дж}/(\text{кг} \cdot ^\circ\text{C})$; удельная теплота плавления льда равна $\lambda = 3,3 \cdot 10^5 \text{ Дж}/\text{кг}$. Ускорение свободного падения вблизи поверхности земли $g = 9,8 \text{ м}/\text{с}^2$. Ответ дайте в километрах и округлите до целого.

Решение

Перед началом падения градины ее энергия полностью была потенциальной и равнялась $E = mgh$, где m — масса градины, h — искомая высота. При падении известная доля энергии ушла на нагрев окружающей среды, а остальная в конечном счете превратилась в тепловую энергию градины (частично до, частично в процессе столкновения с землей). При этом тепловая энергия, необходимая для нагрева и плавления градины задается выражением:

$$Q = cm\Delta t + \lambda m.$$

Приравняв $Q = (1 - \alpha)E$, получим:

$$c\Delta t + \lambda = (1 - \alpha)gh \Rightarrow h = \frac{c\Delta t + \lambda}{(1 - \alpha)g} \approx 42 \text{ км}.$$

Реально облака, содержащие градины, формируются на значительно меньшей высоте, поэтому рассмотренный процесс не происходит на практике.

Ответ: 42 ± 1 .

Вторая попытка. Задачи 10–11 класса

Задача II.2.4.1. Взлет паука (15 баллов)

Темы: напряженность электрического поля.

Условие

Маленький паук размером 1 мм и массой m мг может подняться в воздух за счет электрического заряда на нитях своей паутины q нКл. Между землей и облаками постоянно существует электрическое поле, напряженность которого у поверхности земли на расстояниях, сравнимых с размером паука, может достигать больших значений. Найти минимальное значение напряженности электрического поля E_0 , при котором паук сможет взлететь. Ускорение свободного падения $g = 9,8 \text{ м/с}^2$. Ответ дать в кВ/м с точностью 1 кВ/м.

Решение

Условие взлета паука $mg = qE_0$.

Отсюда находим минимальное значение напряженности электрического поля:

$$E_0 = \frac{mg}{q}.$$

Погрешность 1 кВ/м.

Диапазоны

Величина	min	max	Шаг
m	3,0	7,0	0,5
q	0,9	1,3	0,1

Ответ: $\frac{mg}{q}$.

Задача II.2.4.2. Скорость заполнения ведра (20 баллов)

Темы: расход вещества, равномерное движение.

Условие

Десятиклассник наливает воду в пластмассовое ведро из шланга сечением $S = 5,0 \text{ см}^2$. Ведро имеет форму усеченного конуса высотой $h = 26 \text{ см}$, нижним диаметром $d_1 = 18 \text{ см}$ и верхним диаметром $d_2 = 25 \text{ см}$. В какой-то момент времени уровень воды в ведре имеет высоту $z = 12 \text{ см}$. Найти скорость подъема уровня воды в этот момент времени. Скорость струи воды, вытекающей из шланга, равна $v = 10 \text{ см/с}$. Ответ дать в мм/с с точностью 0,1 мм/с.

Решение

Так как вода в ведро поступает равномерно, то постоянный расход воды $Q = S\nu$ будет равен произведению площади горизонтального сечения ведра на уровне воды на скорость подъема уровня воды в этот момент времени:

$$S\nu = \frac{\pi d^2}{4}u,$$

где d — диаметр горизонтального сечения на высоте z . Из геометрии усеченного конуса следует, что $d = d_1 + (d_2 - d_1)z/h$. Отсюда находим скорость подъема уровня воды:

$$u = \frac{4S\nu}{\pi(d_1 + (d_2 - d_1)z/h)^2} = 1,4 \text{ мм/с.}$$

Погрешность 0,1 мм/с.

Ответ: $1,4 \pm 0,1$.

Задача II.2.4.3. Цикл Отто (20 баллов)

Темы: тепловая машина.

Условие

В бензиновом двигателе внутреннего сгорания расход топлива равен $q = 0,65 \cdot 10^{-3}$ кг/с. Удельная теплота сгорания бензина $r = 46$ МДж/кг. За время t минут все отданное двигателем тепло $Q = 12,2$ МДж получает система охлаждения двигателя. Рабочий процесс такого двигателя можно приближенно описать идеальным циклом Отто, состоящим из двух изохор и двух адиабат. КПД цикла Отто рассчитывается по формуле $\eta = 1 - 1/n^{\gamma-1}$, где γ — показатель адиабаты сгорающей топливо-воздушной смеси. Найти степень сжатия двигателя $n = V_2/V_1$, где V_2 — максимальный и V_1 — минимальный объемы смеси. Ответ дать с точностью 0,1.

Решение

От нагревателя получается тепло qrt и отдается холодильнику тепло Q . Тогда КПД тепловой машины приравняем КПД цикла Отто:

$$1 - \frac{Q}{qrt} = 1 - \frac{1}{n^{\gamma-1}}.$$

Отсюда находим степень сжатия двигателя:

$$n = \left(\frac{qrt}{Q} \right)^{\frac{1}{\gamma-1}}.$$

Диапазоны

Величина	min	max	Шаг
t	9,5	10,5	0,1
γ	1,30	1,37	0,01

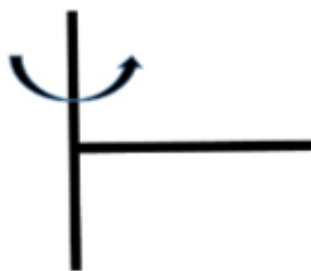
Ответ: $\left(\frac{qrt}{Q} \right)^{\frac{1}{\gamma-1}}$.

Задача П.2.4.4. Энергия вращающегося прута (25 баллов)

Темы: равноускоренное вращение, кинетическая энергия.

Условие

Два тонких стальных прута соединили под прямым углом так, что к середине первого прута приварен конец второго прута. Первый прут расположили вдоль вертикальной оси, а второй прут начинают вращать в горизонтальной плоскости с угловой скоростью $\omega = \beta t$, где угловое ускорение $\beta = 1,1 \text{ рад/с}^2$. Найти кинетическую энергию второго прута в момент времени $t = 3,0 \text{ с}$. Масса второго прута $m \text{ кг}$, а его длина $L \text{ м}$. Ответ дать в Дж с точностью 0,1 Дж.



Решение

Мысленно разрежем прут на небольшие кусочки длиной Δl . Такой кусочек массой $\frac{m\Delta l}{L}$ имеет скорость $V = \beta tl$, где l — расстояние от оси вращения. Кинетическая энергия кусочка равна $\Delta E = \frac{m\Delta l}{L} \frac{\omega^2 l^2}{2}$. Сложив все эти энергии, получим полную кинетическую энергию второго прута $\left(\sum_0^L l^2 \Delta l = \frac{L^3}{3} \right)$:

$$E = \frac{m\beta^2 t^2 L^2}{6}.$$

Погрешность 0,2 Дж.

Диапазоны

Величина	min	max	Шаг
m	3,0	7,0	0,5
L	1,3	1,7	0,1

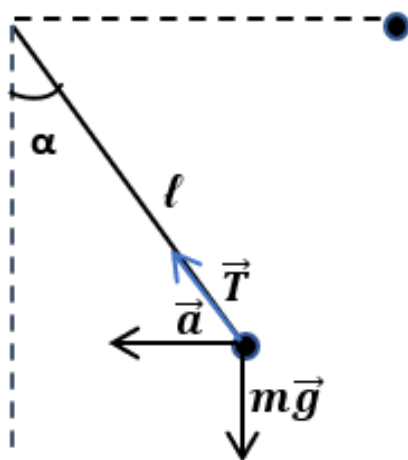
Ответ: $\frac{m\beta^2 t^2 L^2}{6}$.

Задача II.2.4.5. (20 баллов)

Темы: динамика.

Условие

К стене прикрепили тонкую невесомую нить длиной 26 см с грузом массой 500 г на втором конце. Подвешенный груз отклонили от стены так, что нить приняла горизонтальное положение и отпустили (см. рисунок). Определить, на каком расстоянии от стены ускорение груза будет направлено горизонтально? Ответ выразить в см и округлите до целых.



Решение

$$\text{ЗСЭ: } mgl = mgl(1 - \cos \alpha) + \frac{mv^2}{2} \Rightarrow gl \cos \alpha = \frac{v^2}{2}.$$

$$\text{II з-н на нормаль: } T - mg \cos \alpha = m \frac{v^2}{l} \Rightarrow T = 3mg \cos \alpha.$$

$$\text{II з-н на } OY: T \cos \alpha = mg \Rightarrow 3 \cos^2 \alpha = 1 \Rightarrow \cos \alpha = \frac{1}{\sqrt{3}}.$$

$$\text{Или } \operatorname{tg} \alpha = \frac{\sqrt{1 - \cos^2 \alpha}}{\cos \alpha} = \sqrt{2}, \alpha \approx 55^\circ.$$

Из тригонометрической функции $\cos \alpha$ найдем расстояние от груза до стены, зная угол и гипотенузу.

Ответ: 15.

Третья попытка. Задачи 10–11 класса

Задача II.2.6.1. Два прута (15 баллов)

Темы: момент силы.

Условие

Два тонких стальных прута соединили под прямым углом так, что к середине первого прута приварен конец второго прута. Первый прут закрепили неподвижно в вертикальном положении. Найти результирующий момент всех вертикальных сил, действующих на второй прут, относительно его свободного конца. Масса второго прута m кг, а его длина L м. Ускорение свободного падения $g = 9,8 \text{ м/с}^2$. Ответ дать в $\text{Н} \cdot \text{м}$ с точностью $1 \text{ Н} \cdot \text{м}$.



Решение

На второй прут действуют две вертикальные силы: направленная вниз сила тяжести и направленная вверх сила реакции первого прута. Их суммарный момент сил равен:

$$N = mgL/2.$$

Погрешность $1 \text{ Н} \cdot \text{м}$.

Диапазоны

Величина	min	max	Шаг
m	3,0	7,0	0,5
L	1,3	1,7	0,1

Ответ: $N = mgL/2$.

Задача II.2.6.2. Время заполнения ведра (20 баллов)

Темы: расход вещества, равномерное движение.

Условие

Десятиклассник наливает воду в пластмассовое ведро из шланга сечением S . Ведро имеет форму усеченного конуса высотой $h = 26$ см, нижним диаметром $d_1 = 18$ см и верхним диаметром $d_2 = 25$ см. За какое время пустое ведро заполнится полностью? Скорость струи воды, вытекающей из шланга, равна v см/с. Ответ дать в с с точностью 1 с.

Решение

Так как вода в ведро поступает равномерно, то объем ведра равен произведению расхода воды $Q = Sv$ на время заполнения t_1 :

$$V = Svt_1.$$

Отсюда находим время заполнения ведра водой:

$$t = \frac{\pi h(d_1^2 + d_1d_2 + d_2^2)}{12Sv}.$$

Погрешность 2 с.

Диапазоны

Величина	min	max	Шаг
m	3,0	7,0	0,5
L	1,3	1,7	0,1

Ответ: $\frac{\pi h(d_1^2 + d_1d_2 + d_2^2)}{12Sv}$.

Задача II.2.6.3. Система охлаждения двигателя (20 баллов)

Темы: расход жидкости, тепловой баланс.

Условие

В бензиновом двигателе внутреннего сгорания за время t мин все отданное работающим двигателем тепло Q МДж получает система охлаждения двигателя. В системе охлаждения двигателя циркулирует жидкость — антифриз, который прокачивается насосом, забирает тепло у рубашки двигателя и отдает это тепло радиатору. Рубашка и радиатор соединяются трубками (их площадь сечения $S = 3,0$ см²). В рабочем режиме двигателя разность температур антифриза на входе и выходе рубашки равна $\Delta T = 10$ °С. Найти скорость течения антифриза V по трубкам. Плотность антифриза $\rho = 1120$ кг/м³, а его удельная теплоемкость $C = 3600$ Дж/(кг · °С). Ответ дать в м/с с точностью 0,1 м/с.

Решение

За небольшой промежуток времени τ в рубашку втекает масса антифриза $\rho SV\tau$. Тогда запишем уравнение теплового баланса:

$$\frac{Q\tau}{t} = \rho SV\tau C\Delta T.$$

Отсюда находим скорость течения антифриза

$$V = \frac{Q}{t\rho SC\Delta T}.$$

Погрешность 0,2 м/с.

Диапазоны

Величина	min	max	Шаг
t	9,5	10,5	0,1
Q	11,5	12,7	0,01

Ответ: $\frac{Q}{t\rho SC\Delta T}$.

Задача II.2.6.4. Высота полета (25 баллов)

Темы: второй закон Ньютона, ускорение, скорость.

Условие

Маленький паук размером 1 мм и массой $m = 5,0$ мг может подняться в воздух за счет электрического заряда на нитях своей паутины $q = 1,0$ нКл. Между землей и облаками постоянно существует электрическое поле, напряженность ($E = 53$ кВ/м) которого у поверхности земли на расстояниях, сравнимых с размером паука, может достигать больших значений. Известно, что на паука в полете также действует сила сопротивления воздуха $F_c = -kV$, где коэффициент сопротивления равен $k = 50$ мкН/(м/с). На какую высоту H поднимется паук за время τ с? Ускорение свободного падения $g = 9,8$ м/с². Ответ дать в см с точностью 0,1 см.

Решение

Уравнение движения паука:

$$ma = qE - mg - kV.$$

Скорость максимальна при нулевом ускорении $a = 0$:

$$V_m = \frac{qE - mg}{k}.$$

Для небольшого промежутка времени Δt :

$$m \frac{\Delta V}{\Delta t} = qE - mg - k \frac{\Delta h}{\Delta t}.$$

Отсюда $m\Delta V = (qE - mg)\Delta t - k\Delta h$.

Суммируя это уравнение по Δt от 0 до τ получим высоту:

$$H = V_m \left(\tau - \frac{m}{k} \right).$$

Погрешность 0,1 см.

Диапазоны

Величина	min	max	Шаг
τ	0,30	0,60	0,01

Ответ: $V_m \left(\tau - \frac{m}{k} \right)$.

Задача II.2.6.5. (20 баллов)

Темы: электростатика.

Условие

На краях диэлектрической спицы длиной 10 см закреплены заряженные шарики: слева закреплен шарик с зарядом +9 нКл, а справа — с зарядом +16 нКл. Между шариками на спице расположена бусинка, способная скользить по ней практически без трений.

Определить на каком расстоянии до левого шарика окажется бусина через очень большое время, если сообщить ей заряд -2 нКл. Ответ выразить в мм и округлить до целых.

Решение

Условие равновесия среднего шарика: $\frac{kq_1q_3}{r_1^2} = \frac{kq_2q_3}{r_2^2}$.

Кроме того $r_1 + r_2 = l$.

Решая эту систему, получим $r_1 = \frac{l\sqrt{q_1}}{\sqrt{q_2} + \sqrt{q_1}} = 43$ мм.

Ответ: 43.

Четвертая попытка. Задачи 10–11 класса

Задача П.2.8.1. Бензиновый двигатель (15 баллов)

Темы: КПД.

Условие

В бензиновом двигателе внутреннего сгорания расход топлива равен $q = 0,65 \cdot 10^{-3}$ кг/с. Удельная теплота сгорания бензина $r = 46$ МДж/кг. За время t мин все отданное двигателем тепло Q МДж получает система охлаждения двигателя. Найти КПД бензинового двигателя. Ответ дать в % с точностью 1%.

Решение

От нагревателя получается тепло qrt и отдается холодильнику тепло Q . Тогда:

$$\text{КПД} = \left(1 - \frac{Q}{qrt}\right) \times 100\%.$$

Погрешность 1%.

Диапазоны

Величина	min	max	Шаг
t	9,5	10,5	0,1
Q	11,5	12,7	0,1

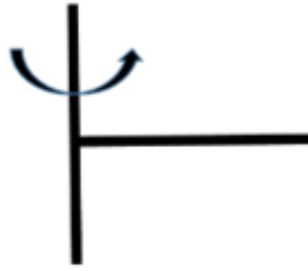
Ответ: $\left(1 - \frac{Q}{qrt}\right) \times 100\%$.

Задача П.2.8.2. Полное ускорение (20 баллов)

Темы: равноускоренное вращение.

Условие

Два тонких стальных прута соединили под прямым углом так, что к середине первого прута приварен конец второго прута. Первый прут расположили вдоль вертикальной оси, а второй прут начинают вращать в горизонтальной плоскости с угловой скоростью $\omega = \beta t$, где угловое ускорение $\beta = 1,1$ рад/с². Найти полное ускорение свободного конца второго прута в момент времени t с. Масса второго прута $m = 5,0$ кг, а его длина L м. Ускорение свободного падения $g = 9,8$ м/с². Ответ дать в м/с² с точностью 0,1 м/с².



Решение

Скорость свободного конца второго прута равна $V = \beta tL$, а его центростремительное ускорение это $\beta^2 t^2 L$. Так как свободный конец второго прута движется равноускоренно по окружности, то составляющая ускорения, касательная к окружности, равна βL . Полное ускорение находим по теореме Пифагора:

$$a = \sqrt{(\beta^2 t^2 L)^2 + (\beta L)^2} = \beta L \sqrt{1 + \beta^2 t^4}.$$

Погрешность $0,1 \text{ м/с}^2$.

Диапазоны

Величина	min	max	Шаг
L	1,3	1,7	0,1
t	2,8	3,9	0,1

Ответ: $\beta L \sqrt{1 + \beta^2 t^4}$.

Задача II.2.8.3. Максимальная скорость полета (20 баллов)

Темы: второй закон Ньютона, сила сопротивления.

Условие

Маленький паук размером 1 мм и массой $m = 5,0 \text{ мг}$ может подняться в воздух за счет электрического заряда на нитях своей паутины $q = 1,0 \text{ нКл}$. Между землей и облаками постоянно существует электрическое поле, напряженность ($E = 53 \text{ кВ/м}$) которого у поверхности земли на расстояниях, сравнимых с размером паука, может достигать больших значений. Известно, что на паука в полете также действует сила сопротивления воздуха $F_c = -kV$, где коэффициент сопротивления равен $k = 50 \text{ мкН/(м/с)}$. Найти максимальное значение скорости полета паука V_m . Ускорение свободного падения $g = 9,8 \text{ м/с}^2$. Ответ дать в см/с с точностью $0,1 \text{ см/с}$.

Решение

Уравнение движения паука:

$$ma = qE - mg - kV.$$

Скорость максимальна при нулевом ускорении $a = 0$:

$$V_M = \frac{qE - mg}{k} = 8,0 \text{ см/с.}$$

Погрешность 0,1 см/с.

Ответ: $8,0 \pm 0,1$.

Задача II.2.8.4. Время выливания воды (25 баллов)

Темы: расход вещества, равноускоренное движение.

Условие

Десятиклассник выливает воду из полного пластмассового ведра таким образом, что расход воды уменьшается со временем t линейным образом: $Q = q_0 \cdot (1 - t/t_2)$, где q_0 л/с — начальный расход, t_2 — время выливания всей воды из ведра ($0 \leq t \leq t_0$). Расход воды — это объем воды, выливаемый из ведра за единицу времени. Ведро имеет форму усеченного конуса высотой h см, нижним диаметром $d_1 = 18$ см и верхним диаметром $d_2 = 25$ см. Найти t_2 . Ответ дать в с с точностью 0,1 с.

Решение

Так как вода из ведра выливается равноускорено, то объем ведра равен произведению среднего расхода воды $Q_0/2$ на время выливания t_2 :

$$V = t_2 Q_0 / 2.$$

Отсюда находим время выливания воды:

$$t_2 = \frac{\pi h (d_1^2 + d_1 d_2 + d_2^2)}{6Q_0}.$$

Погрешность 0,2 с.

Диапазоны

Величина	min	max	Шаг
Q_0	1,0	6,0	0,5
h	22	32	1

Ответ: $\frac{\pi h (d_1^2 + d_1 d_2 + d_2^2)}{6Q_0}$.

Задача II.2.8.5. (20 баллов)

Темы: равноускоренное движение.

Условие

На представлении жонглер одной рукой подбрасывает мяч и в этот же момент ловит следующий. Всего жонглер использует пять мячей. Определите, чему равно расстояние от мяча в наивысшей точке до соседних мячей, если жонглер подбрасывает их на 1,25 м? Ответ выразите в см и округлите до целых. Ускорение свободного падения считайте равным 10 м/с^2 .

Решение

$$t_{\text{н}} = \frac{v_0}{g};$$

$$\Delta t = \frac{2t_{\text{н}}}{4} = \frac{v_0}{2g}.$$

$$\Delta y = h_{\text{н}} - v_0 \Delta t + \frac{g \Delta t^2}{2} = \frac{v_0^2}{2g} - \frac{v_0^2}{2g} + \frac{g}{2} \frac{v_0^2}{4g^2} = \frac{v_0^2}{8g} \approx 31 \text{ см}.$$

Ответ: 31.

Автономные транспортные системы

Первый отборочный этап

Инженерный тур

Задача П.3.1. Определение светового сигнала автомобиля (5 баллов)

Темы: компьютерное зрение, анализ цветовой гистограммы изображения.

Условие

На изображении представлен автомобиль или его фрагмент. Ваша задача написать функцию, определяющую, какой из двух сигналов автомобиль подаёт: сигнал поворота или стоп-сигнал. На каждом изображении только один сигнал. Сигнал красного цвета — стоп-сигнал, сигнал жёлтого цвета — сигнал поворота.

Для вас подготовлены несколько файлов «.py», набор тестовых изображений и аннотации к нему. Среди подготовленных файлов есть:

- `eval.py` — файл с функцией распознавания сигнала автомобиля. Именно эту функцию вам необходимо дописать!
- `main.py` — файл, проверяющий точность работы вашего алгоритма. Не редактируйте его. `main.py` использует написанные вами функции из `eval.py` и сверяет истинные метки с предсказанием вашего алгоритма.
- `annotations.csv` — файл, устанавливающий соотношение между изображениями с сигналами автомобиля на них. В каждой строке файла содержится путь к файлу с изображением и название сигнала.

Материалы задачи для скачивания: <https://disk.yandex.ru/d/Y4Ub7rG0C2mCow>.

Отладьте работу своего алгоритма на локальной машине и добейтесь точности, устраивающей вас. После этого присылайте решение на онлайн-платформу. Изображения, используемые для тестирования решений на онлайн-платформе, не совпадают с тестовыми изображениями, предоставленными вам.

В качестве решения, пришлите отредактированный файл `eval.py`. Либо архив `*.zip` с файлом `eval.py` и остальными файлами, требующимися для его работы.

Технические ограничения

Размер решения ограничен: не более 0,01 МБ. Если ваш алгоритм успешно проверен платформой, то следующее решение можно прислать только через 10 минут. Если ваш алгоритм в ходе проверки выдал сообщение об ошибке, то следующее решение можно прислать сразу.

Пакеты, ориентированные на работу с изображениями и данными, используемые на онлайн-платформе: Python 3.8.10.

```
dlib 19.24.0; keras 2.8.0; Keras-Preprocessing 1.1.2; imutils 0.5.4;
numpy 1.22.4; opencv-python 4.6.0.66; pandas 1.4.3; scikit-image 0.19.3;
scikit-learn 1.1.1; scipy 1.8.1; tensorflow-cpu 2.8.2.
```

Используйте совместимые пакеты.

Решение

Материалы задачи, решение и тесты: <https://disk.yandex.ru/d/Y4Ub7rG0C2mCow>.

Решить задачу можно множеством способов. Ниже приведён вариант с подсчётом количества красных и жёлтых пикселей на изображении. Если больше красных пикселей, то на изображении стоп-сигнал, иначе — сигнал поворота. Определение пороговых значений, составляющих цвета пикселей для красного и жёлтого цвета, необходимо подобрать путём экспериментов. Подбирать пороги можно в различных цветовых пространствах. В приведённом решении используется цветовое пространство по умолчанию — BGR.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1  # -*- coding: utf-8 -*-
2  import cv2
3  import numpy as np
4
5  def predict_car_signal(image):
6      """
7      Функция, определяющая сигнал автомобиля на изображении
8      Входные данные: изображение (bgr)
9      Выходные данные: строка с названием сигнала
10
11     Примеры вывода: "stop signal"
12                    "turn signal"
13     """
14     image = cv2.resize(image, (400, 300))
15
16     red = cv2.inRange(image, (0, 0, 150), (120, 90, 255))
17     yellow = cv2.inRange(image, (0, 80, 200), (120, 255, 255))
18     orange_sum = np.sum(yellow)
19     red_sum = np.sum(red)
20
21     if orange_sum > red_sum:
22         car_signal = "turn signal"
23     else:
24         car_signal = "stop signal"
25
26     return car_signal
```

Задача II.3.2. Распознавание свободных парковочных мест (15 баллов)

Темы: компьютерное зрение, анализ градиента цвета на изображении.

Условие

На изображении представлены три парковочных места. Ваша задача написать функцию, определяющую номера свободных парковочных мест и выводящую их список в порядке возрастания. Места пронумерованы сверху вниз, нумерация начинается с 1.

Для вас подготовлены несколько файлов «.py», набор изображений и аннотации к нему. Среди подготовленных файлов есть:

- `eval.py` — файл с функцией определения свободных парковочных мест. Именно эту функцию вам необходимо дописать!
- `main.py` — файл, проверяющий точность работы вашего алгоритма. Не редактируйте его. `main.py` использует написанные вами функции из `eval.py` и сверяет истинные метки с предсказанием вашего алгоритма.
- `annotations.csv` — файл, устанавливающий соотношение между изображениями и свободными парковочными местами на них. В каждой строке файла содержится путь к файлу и список номеров свободных парковочных мест.

Материалы задачи для скачивания: <https://disk.yandex.ru/d/Dlw18stoFRitGw>.

Отладьте работу своего алгоритма на локальной машине и добейтесь точности, устраивающей вас. После этого присылайте решение на онлайн-платформу. Изображения, используемые для тестирования решений на онлайн-платформе, не совпадают с тестовыми изображениями, предоставленными вам.

В качестве решения пришлите отредактированный файл `eval.py`. Либо архив `*.zip` с файлом `eval.py` и остальными файлами, требующимися для его работы.

Технические ограничения

Размер решения ограничен: не более 0,01 МБ. Если ваш алгоритм успешно проверен платформой, то следующее решение можно прислать только через 10 минут. Если ваш алгоритм в ходе проверки выдал сообщение об ошибке, то следующее решение можно прислать сразу.

Пакеты, ориентированные на работу с изображениями и данными, используемые на платформе проверки: Python 3.8.10.

```
dlib 19.24.0; keras 2.8.0; Keras-Preprocessing 1.1.2; imutils 0.5.4;
numpy 1.22.4; opencv-python 4.6.0.66; pandas 1.4.3; scikit-image 0.19.3;
scikit-learn 1.1.1; scipy 1.8.1; tensorflow-cpu 2.8.2.
```

Используйте совместимые пакеты.

Решение

Материалы задачи, решение и тесты: <https://disk.yandex.ru/d/LV6XN5LitHL0bw>.

Ограничение размера решения вынуждает решать задачу без использования нейронных сетей. Пустое парковочное место можно отличить от автомобиля по текстуре асфальта. Она имеет зернистую структуру из относительно небольших областей одинакового цвета. На такой текстуре градиент цвета от пикселя к пикселю изменяется скачкообразно и гораздо чаще, чем на изображении автомобиля. В представленном решении изображение делится на три области, соответствующие парковочным местам. Для каждой области вычисляется НОГ-гистограмма, она отражает частоту и величину изменения цвета на изображении. Если значение градиента выше определённого порога, то на изображении асфальт, соответственно, парковочное место свободно.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1  # -*- coding: utf-8 -*-
2  import cv2
3  import numpy as np
4  from skimage.feature import hog
5
6  def predict_number_empty_parking_places(image):
7      """
8      Функция, определяющая номера свободных парковочных мест на изображении
9      Входные данные: изображение (bgr)
10     Выходные данные: список номеров свободных парковочных мест, в порядке
    ↪     возрастания
11
12     Примеры вывода: [1, 2, 3]
13                    []
14                    [1]
15                    [2, 3]
16     """
17
18     parking_places = [image[20:200, 20:400],
19                      image[220:400, 20:400],
20                      image[420:600, 20:400]]
21     result = []
22     for i, parking_place in enumerate(parking_places):
23         fd, hog_image = hog(parking_place, orientations=9, pixels_per_cell=(8, 8),
24                             cells_per_block=(2, 2), visualize=True,
25                             ↪     multichannel=True)
26         if np.max(fd) < 0.4:
27             result.append(i+1)
28     return result
```

Задача II.3.3. Детектирование и распознавание знаков дорожного движения (35 баллов)

Темы: компьютерное зрение, детектирование, бинаризация, контурный анализ, сравнение с шаблоном.

Условие

На изображении представлен знак дорожного движения. Ваша задача написать функцию, определяющую, где на изображении расположен знак дорожного движения и как он называется. Возможных вариантов знаков пять: NoDrive, Stop, Parking, RoadWorks, PedestrianCrossing. На каждом изображении только один знак дорожного движения.

Для вас подготовлены несколько файлов «.ру», набор изображений знаков и аннотации к нему. Среди подготовленных файлов есть:

- `eval.py` — файл с ключевыми функциями: предобработки изображений, загрузки эталонных изображений, детектирования и распознавания знака. Именно эти функции Вам необходимо дописать!

-
- `main.py` — файл, проверяющий точность работы вашего алгоритма. Не редактируйте его. `main.py` использует написанные вами функции из `eval.py` и светит истинные метки с предсказанием вашего алгоритма.
 - `annotations.csv` — файл, устанавливающий соотношение между изображениями, координатами знаков и их названиями. В каждой строке файла содержится путь к файлу с изображением, название знака и координаты ограничивающей знак рамки.

Материалы задачи для скачивания: https://disk.yandex.ru/d/32fq930Fqwi_Zw.

Отладьте работу своего алгоритма на локальной машине и добейтесь точности, устраивающей вас. После этого присылайте решение на онлайн-платформу. Изображения, используемые для тестирования решений на онлайн-платформе, не совпадают с тестовыми изображениями, предоставленными вам.

В качестве решения, пришлите отредактированный файл `eval.py`. Либо архив `*.zip` с файлом `eval.py` и остальными файлами, требующимися для его работы.

Технические ограничения

Размер решения ограничен: не более 30 МБ. Если ваш алгоритм успешно проверен платформой, то следующее решение можно прислать только через 10 минут. Если ваш алгоритм в ходе проверки выдал сообщение об ошибке, то следующее решение можно прислать сразу.

Пакеты, ориентированные на работу с изображениями и данными, используемые на платформе проверки: Python 3.8.10.

```
dlib 19.24.0; keras 2.8.0; Keras-Preprocessing 1.1.2; imutils 0.5.4;
numpy 1.22.4; opencv-python 4.6.0.66; pandas 1.4.3; scikit-image 0.19.3;
scikit-learn 1.1.1; scipy 1.8.1; tensorflow-cpu 2.8.2.
```

Используйте совместимые пакеты.

Решение

Материалы задачи, решение и тесты: <https://disk.yandex.ru/d/wCWAJppRNxbavQ>.

Для детектирования дорожных знаков можно применить множество алгоритмов детектирования объектов, включая нейросетевые.

В приведённом решении используется детектирование объекта по цвету и распознавание сравнением с шаблоном. Исходное изображение бинаризуется по нескольким экспериментально подобранным порогам. Для знаков различных цветов пороги разные. На чёрно-белом изображении проводится поиск контуров. Самый большой по площади контур считается детектированным объектом. Детектированный объект масштабируется до размеров чёрно-белых эталонных изображений знаков и сравнивается с ними попиксельно. Если результат сравнения превышает некий порог, то объект считается успешно распознанным.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1  # -*- coding: utf-8 -*-
2  from pathlib import Path
3  import cv2 as cv
4  import numpy as np
5
6  blue_bottom, blue_up = (0, 65, 50), (20, 255, 255)
7  red_bottom, red_up = (105, 52, 98), (170, 255, 255)
8
9  def find_color_contours(mask):
10     mask = cv.dilate(mask, np.ones((2, 2), np.uint8))
11     mask = cv.erode(mask, np.ones((2, 2), np.uint8))
12
13     contours, _ = cv.findContours(mask, cv.RETR_EXTERNAL, cv.CHAIN_APPROX_NONE)
14
15     if contours:
16         # cnt_image = cv.drawContours(mask.copy(), contours, -1, 255)
17         # cv.imshow('cnts', cnt_image)
18         contours = sorted(contours, key=cv.contourArea, reverse=True)
19         if cv.contourArea(contours[0]) > 100:
20             x, y, w, h = cv.boundingRect(contours[0])
21             return [cv.resize(mask[y:y + h, x:x + w], (64, 64)), (x, y, x + w, y +
↪ h)]
22
23     return [np.zeros((1, 1)), (0, 0, 0, 0)]
24
25 def load_etalon_list():
26     """ Функция осуществляет загрузку изображений эталонных знаков.
27     Вы можете провести бинаризацию эталонных изображений здесь же. В этом случае
28     верните список бинаризованных изображений.
29     Если вы не собираетесь использовать эту функцию, пусть возвращает пустой
↪ список []
30     """
31
32     control_stop = cv.inRange(cv.cvtColor(cv.imread('etalons/Stop.jpg'),
↪ cv.COLOR_BGR2HSV),
33                               red_bottom, red_up)
34     control_stop = cv.resize(control_stop, (64, 64))
35
36     control_noDrive = cv.inRange(cv.cvtColor(cv.imread('etalons/NoDrive.jpg'),
↪ cv.COLOR_BGR2HSV),
37                                  red_bottom, red_up)
38     control_noDrive = find_color_contours(control_noDrive)[0]
39
40     control_roadWorks = cv.inRange(cv.cvtColor(cv.imread('etalons/RoadWorks.jpg'),
↪ cv.COLOR_BGR2HSV),
41                                    red_bottom, red_up)
42     control_roadWorks = find_color_contours(control_roadWorks)[0]
43
44     control_parking = cv.inRange(cv.cvtColor(cv.imread('etalons/Parking.jpg'),
↪ cv.COLOR_BGR2HSV), blue_bottom, blue_up)
45     control_parking = find_color_contours(control_parking)[0]
46
47     control_pedestrian =
48     ↪ cv.inRange(cv.cvtColor(cv.imread('etalons/PedestrianCrossing.jpg'),
49     ↪ cv.COLOR_BGR2HSV),
50                               blue_bottom, blue_up)
51     control_pedestrian = cv.resize(control_pedestrian, (64, 64))
52
53     etalon_list = {'PedestrianCrossing': control_pedestrian,
54                   'Stop': control_stop,

```

```

53         'NoDrive': control_noDrive,
54         'Parking': control_parking,
55         'RoadWorks': control_roadWorks}
56
57     return etalon_list
58
59 def predict_label_and_box(img, etalon_list):
60     """
61     Функция, детектирующая знаки и распознающая их
62     Входные данные: изображение (bgr), список изображений эталонных знаков
63     Выходные данные: список с метками объектов и координатами ограничивающих
↪     рамок
64
65     Формат вывода: [[label, (x, y, x2, y2)]]
66                     label - текстовая метка с названием знака
67                     x и y - координаты левого верхнего угла ограничивающей рамки
68                     x2 и y2 - координаты правого нижнего угла ограничивающей рамки
69
70     Примеры вывода: [{"PedestrianCrossing", (10, 14, 18, 25)},
71                     {"Parking", (110, 114, 118, 215)}] - если обнаружили два
↪     знака
72
73                     [{"RoadWorks", (10, 11, 18, 15)}] - если обнаружили один знак
74
75                     [] - если не обнаружили ни одного знака
76     """
77
78     hsv_img = cv.cvtColor(img, cv.COLOR_BGR2HSV)
79
80     blue_mask = cv.inRange(hsv_img, blue_bottom, blue_up)
81     red_mask = cv.inRange(hsv_img, red_bottom, red_up)
82
83     blue_mini_mask = find_color_contours(blue_mask)
84     red_mini_mask = find_color_contours(red_mask)
85
86     predicted_label_and_box = []
87
88     ans = [(0, " ")]
89
90     z = np.zeros((64, 64))
91     if blue_mini_mask[0].any():
92
93         for i in ('PedestrianCrossing', 'Parking'):
94             z[etalon_list[i] == blue_mini_mask[0]] = 1
95             value = np.sum(z)
96             print(i, value)
97             if value > 1700:
98                 ans.append((value, [i, blue_mini_mask[1]]))
99             z = np.zeros((64, 64))
100
101     if red_mini_mask[0].any():
102         for i in ('Stop', 'NoDrive', 'RoadWorks'):
103             z[etalon_list[i] == red_mini_mask[0]] = 1
104             value = np.sum(z)
105             print(i, value)
106             if value > 1700:
107                 ans.append((value, [i, red_mini_mask[1]]))
108             z = np.zeros((64, 64))
109
110     ans = sorted(ans, key=lambda x: x[0], reverse=True)

```

```
111     if ans[0][0]:
112         predicted_label_and_box.append(ans[0][1])
113
114     return predicted_label_and_box
```

Задача II.3.4. Распознавание цветовой маркировки груза (45 баллов)

Темы: компьютерное зрение, детектирование, HOG-SVM детекторы, бинаризация.

Условие

На изображении представлен груз с цветовой маркировкой. Маркировка состоит из трёх полос. Цвет полосы кодирует символ: синий — «В», фиолетовый — «М», зелёный — «G», чёрный — «0». Таким образом, маркировка может быть расшифрована как последовательность из трёх символов. Маркировка читается слева направо. Первая — левая линия вместо чёрного цвета — всегда имеет жёлтый цвет, кодирующий символ — «0». Ваша задача написать функцию, расшифровывающую цветовую маркировку на грузе. На каждом изображении только один груз. Линии маркировки всегда ориентированы вертикально.

Для вас подготовлены несколько файлов «.py», набор изображений и аннотации к нему. Среди подготовленных файлов есть:

- `eval.py` — файл с функцией расшифровки цветовой маркировки. Именно эту функцию Вам необходимо дописать!
- `main.py` — файл, проверяющий точность работы вашего алгоритма. Не редактируйте его. `main.py` использует написанные вами функции из `eval.py` и сверяет истинные метки с предсказанием вашего алгоритма.
- `annotations.csv` — файл, устанавливающий соотношение между изображениями и расшифровками цветовых маркировок грузов на них. В каждой строке файла содержится путь к файлу и расшифровка цветовой маркировки.

Материалы задачи для скачивания: <https://disk.yandex.ru/d/wCWAJppRNxbBQ>.

Отладьте работу своего алгоритма на локальной машине и добейтесь точности, устраивающей вас. После этого присылайте решение на онлайн-платформу. Изображения, используемые для тестирования решений на онлайн-платформе, не совпадают с тестовыми изображениями, предоставленными вам.

В качестве решения необходимо прислать отредактированный файл `eval.py`. Либо архив `*.zip` с файлом `eval.py` и остальными файлами, требующимися для его работы.

Технические ограничения

Размер решения ограничен: не более 30 МБ. Если ваш алгоритм успешно проверен платформой, то следующее решение можно прислать только через 10 минут. Если ваш алгоритм в ходе проверки выдал сообщение об ошибке, то следующее решение можно прислать сразу.

Пакеты, ориентированные на работу с изображениями и данными, используемые на платформе проверки: Python 3.8.10.

dlib 19.24.0; keras 2.8.0; Keras-Preprocessing 1.1.2; imutils 0.5.4; numpy 1.22.4; opencv-python 4.6.0.66; pandas 1.4.3; scikit-image 0.19.3; scikit-learn 1.1.1; scipy 1.8.1; tensorflow-cpu 2.8.2.

Используйте совместимые пакеты.

Решение

Материалы задачи, решение и тесты: <https://disk.yandex.ru/d/Txg9i0JbuUILQA>.

Для определения цветовой маркировки грузов можно применить множество комбинаций алгоритмов детектирования и классификации объектов, включая нейросетевые.

В представленном решении детектирование груза и распознавание маркировки выполняются отдельно. Детектирование осуществляется SVM-детектором, обученном на предоставленных тестовых изображениях. Для распознавания цветовой маркировки на детектированном грузе изображение разбивается на области, соответствующие возможным положениям полос. В каждой области подсчитывается число пикселей конкретного цвета. Каких пикселей больше, таков и цвет полосы. Определение пороговых значений составляющих цвета пикселей для различных цветов необходимо подобрать путём экспериментов. Подбирать пороги можно в различных цветовых пространствах. В приведённом решении используется цветовое пространство HSV.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1  # -*- coding: utf-8 -*-
2  import cv2
3  import dlib
4  import numpy as np
5
6  def convert_and_trim_bb(image, rect):
7      # extract the starting and ending (x, y)-coordinates of the
8      # bounding box
9      startX = rect.left()
10     startY = rect.top()
11     endX = rect.right()
12     endY = rect.bottom()
13     # ensure the bounding box coordinates fall within the spatial
14     # dimensions of the image
15     startX = max(0, startX)
16     startY = max(0, startY)
17     endX = min(endX, image.shape[1])
18     endY = min(endY, image.shape[0])
19     # compute the width and height of the bounding box
20     w = endX - startX
21     h = endY - startY
22     # return our bounding box coordinates
23     return startX, startY, w, h
24
25  def load_detector():
26     """
```

```

27     Функция осуществляет загрузку модели машинного обучения из файла.
28     Выходные параметры: список загруженных моделей
29
30     Если вы не собираетесь использовать эту функцию, пусть возвращает пустой
↪ список []
31     Если вы используете несколько моделей, возвращайте их список [detector1,
↪ detector2]
32
33     То, что вы вернёте из этой функции, будет передано вторым аргументом в функцию
↪ predict_box
34     """
35     detectors_list = dlib.simple_object_detector('detector.svm')
36     return detectors_list
37
38 def predict_colors_of_bars(image, detectors_list) -> tuple:
39     """
40     Функция, расшифровывающая цветовую маркировку на грузе
41     Полосы могут быть четырёх цветов
42     (синий - В, фиолетовый - М, зелёный - Г, чёрный - О)
43     Чёрный цвет первой полосы всегда заменяется на жёлтый, обозначение "О" не
↪ меняется.
44
45     Входные данные: изображение (bgr), список загруженных моделей
46     Выходные данные: три символа, обозначающие соответствующим полос маркировки
47
48     Примеры вывода:
49     ВМГ - первая полоса голубая, вторая - фиолетовая, третья - зеленая;
50     ГОВ - первая полоса зеленая, вторая - черная, третья - голубая;
51     ООО - все три полосы черные;
52     """
53
54     rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
55
56     rects = detectors_list(rgb_image)
57     boxes = [convert_and_trim_bb(image, rect) for rect in rects]
58
59     box = boxes[0]
60     x1, y1, w, h = box
61     x2, y2 = x1 + w, y1 + h
62     line_w = int(w / 4)
63
64     box_img = image.copy()[y1:y2, x1:x2]
65     lines = []
66     for i in range(3):
67         lines.append(box_img[:, line_w * i:line_w * (i + 1)])
68
69     result = ''
70     for i, line in enumerate(lines):
71         hsv = cv2.cvtColor(line, cv2.COLOR_BGR2HSV)
72
73         line_black = cv2.inRange(hsv, (0, 0, 0), (179, 50, 50))
74         line_yellow = cv2.inRange(hsv, (40, 100, 60), (65, 255, 255))
75         line_magenta = cv2.inRange(hsv, (150, 100, 60), (179, 255, 255))
76         line_green = cv2.inRange(hsv, (70, 100, 60), (90, 255, 255))
77         line_blue = cv2.inRange(hsv, (90, 100, 50), (110, 255, 255))
78
79         colors = ['black', 'yellow', 'magenta', 'green', 'blue']
80         colors_values = [np.sum(line_black), np.sum(line_yellow),
↪ np.sum(line_magenta), np.sum(line_green),
81                         np.sum(line_blue)]

```

```
82     color_index = colors_values.index(max(colors_values))
83     color = colors[color_index]
84
85     if color in ('black', 'yellow'):
86         result += 'O'
87     elif color == 'magenta':
88         result += 'M'
89     elif color == 'green':
90         result += 'G'
91     elif color == 'blue':
92         result += 'B'
93
94     return result
```