

2. ВТОРОЙ ЭТАП

Задачи второго этапа

3.1. Блок заданий 1

Задача 3.1.1. (5 баллов)

Камеры видеонаблюдения в умном городе записывают видео в чересстрочном (интерлейсном) формате, при чём, для экономии занимаемого объёма памяти цветная составляющая каждого кадра отбрасывается, и каждое изображение записывается на носитель в градациях серого с глубиной 8 бит. Таким образом, каждый пиксель видеопоследовательности представляется одним числом в диапазоне от 0 (полностью чёрный) до 255 (полностью белый). Каждый кадр видеопоследовательности представляет собой матрицу пикселей с шириной W и высотой H пикселей, при чём высота изображения H является чётной.

Интерлейсный формат представления видео подразумевает, что каждый кадр размера $W \times H$ на самом деле хранит два последовательно идущих кадра таким образом, что чётные строки содержат значения яркостей пикселей чётных строк кадра с номером $2K$, а нечётные содержат значения яркостей пикселей нечётных строк кадра с номером $2K + 1$ (K – целое число).

Для воспроизведения чересстрочного видео плеер должен уметь выполнить процесс деинтерлейса, то есть получить из одного интерлейсного кадра размера $W \times H$ два кадра размера $W \times H$. В простейшем случае этот процесс можно разбить на следующие этапы

- разбиение исходного чересстрочного кадра на два полукадра размером $W \times (\frac{H}{2})$, каждый из которых содержит только пиксели одного из двух исходных кадров;
- восстановление недостающих строк путём усреднения значений двух соседних (сверху и снизу) пикселей. Если у восстанавливаемого пикселя отсутствует верхний или нижний сосед, значение этого пикселя копируется из имеющегося соседа. Так как все значения пикселей являются целыми числами, при усреднении дробная часть отбрасывается.

Ваша задача – написать описанный деинтерлейсер.

Формат входных данных

В первой строке дано два целых числа W и H ($16 \leq W \leq 1920, 16 \leq H \leq 1080, H$ - чётное).

В следующих H строках содержится по W чисел $p_{i,j}$ ($0 \leq p_{i,j} \leq 255$), каждое из которых представляет значение j -го пикселя i -ой строки.

Формат выходных данных

Выведите H строк по W значений в каждой – значения пикселей восстановленных строк. Первые $\frac{H}{2}$ строк должны представлять значения пикселей первого кадра, вторые $\frac{H}{2}$ строк должны представлять значения пикселей второго кадра.

Решение

Для решения задачи необходимо сделать ровно то, что сказано в условиях:

Разбить исходный чересстрочный кадр на два полукадра размером $W \times (H/2)$, каждый из которых содержит только пиксели одного из двух исходных кадров; Восстановить недостающие строки путём усреднения значений двух соседних (сверху и снизу) пикселей. Если у восстанавливаемого пикселя отсутствует верхний или нижний сосед, значение этого пикселя копируется из имеющегося соседа. Так как все значения пикселей являются целыми числами, при усреднении дробная часть отбрасывается.

Пример программы-решения

Ниже представлено решение на языке C++

```

1  #include <stdio>
2  #include <stdint>
3
4  using namespace std;
5
6  uint8_t pic[1080][1920]{};
7  int w, h;
8
9  void print_line(uint8_t *buf)
10 {
11     for (int i = 0; i < w; i++)
12         printf("%d ", (int)buf[i]);
13     printf("\n");
14 }
15
16 void deinter_line(uint8_t *up, uint8_t *down)
17 {
18     for (int i = 0; i < w; i++)
19         printf("%d ", (((int)up[i] + down[i]) >> 1));
20     printf("\n");
21 }
22
23 void loop(int y_beg)
24 {
25     for (int y = y_beg; y < h - 2; y += 2) {
26         deinter_line(pic[y], pic[y + 2]);
27     }
28 }
29
30 int main()
31 {
32     scanf("%d%d", &w, &h);
33

```

```

34     for (int y = 0; y < h; y++)
35         for (int x = 0; x < w; x++) {
36             int t;
37             scanf("%d", &t);
38             pic[y][x] = t;
39         }
40
41     loop(0);
42     print_line(pic[h - 2]);
43
44     print_line(pic[1]);
45     loop(1);
46
47     return 0;
48 }

```

Задача 3.1.2. (5 баллов)

Текущая дата на городских табло представляется в виде последовательности из 4 цифр. Однако табло производят две разные фирмы, каждая из которых использует свой формат даты: ДДММ или ММДД, где ММ - номер месяца (от 01 до 12), а ДД - номер дня месяца (от 01 до 31) с ведущими нулями. Гарантируется, что дата хотя бы в одном формате указана верно.

От вас требуется определить, можно ли по заданной последовательности из 4 цифр однозначно определить сегодняшнюю дату.

Формат входных данных

В единственной строке дано четырёхзначное число, представляющее дату в одном из описанных форматов.

Формат выходных данных

YES, если дату можно определить однозначно, и NO в противном случае.

Пример №1

Стандартный ввод
3112
Стандартный вывод
YES

Пример №2

Стандартный ввод
1011
Стандартный вывод
NO

Пример программы-решения

Ниже представлено решение на языке C++

```
1  #include <cstdio>
2  #include <stdint>
3
4  using namespace std;
5
6  uint8_t pic[1080][1920]{};
7  int w, h;
8
9  void print_line(uint8_t *buf)
10 {
11     for (int i = 0; i < w; i++)
12         printf("%d ", (int)buf[i]);
13     printf("\n");
14 }
15
16 void deinter_line(uint8_t *up, uint8_t *down)
17 {
18     for (int i = 0; i < w; i++)
19         printf("%d ", (((int)up[i] + down[i]) >> 1));
20     printf("\n");
21 }
22
23 void loop(int y_beg)
24 {
25     for (int y = y_beg; y < h - 2; y += 2) {
26         deinter_line(pic[y], pic[y + 2]);
27     }
28 }
29
30 int main()
31 {
32     scanf("%d%d", &w, &h);
33
34     for (int y = 0; y < h; y++)
35         for (int x = 0; x < w; x++) {
36             int t;
37             scanf("%d", &t);
38             pic[y][x] = t;
39         }
40
41     loop(0);
42     print_line(pic[h - 2]);
43
44     print_line(pic[1]);
45     loop(1);
46
47     return 0;
48 }
```

Задача 3.1.3. (5 баллов)

Навигационный отдел умного города тестируют новые маршруты общественного транспорта. Однако в базе данных имеется информация только о количестве минут, необходимых на проезд от текущей остановки до следующей. Отделу необходимо знать время проезда от одной до любой другой произвольной остановки. Для этого они посылают в базу данных запрос, представляющий собой пару чисел A и B –

номера остановок одного маршрута, время в пути между которыми они хотят знать. Ваша задача – написать программу, которая будет отвечать на такие запросы.

Формат входных данных

В первой строке дано два целых числа N и M – количество остановок исследуемого маршрута и количество запросов соответственно ($2 \leq N \leq 1000000$, $1 \leq M \leq 1000000$).

Во второй строке дано $N - 1$ целых чисел t_i – время в пути между остановками с номерами i и $i + 1$ ($0 \leq t_i \leq 2^{31} - 1$).

Остановки нумеруются по порядку с 1 до N .

В следующих M строках даны пары целых чисел A и B – номера остановок, между которыми необходимо определить время пути маршрутного транспорта ($1 \leq A, B \leq N$).

Гарантируется, что полное время, затраченное на проезд с начальной до конечной остановок, не превышает $2^{31} - 1$.

Формат выходных данных

В качестве ответа выведите M строк, содержащих ответы на запросы.

Пример №1

Стандартный ввод
5 4
3 4 1 3
1 2
1 3
1 5
2 4
Стандартный вывод
3
7
11
5

Пример программы-решения

Ниже представлено решение на языке C++

```

1  #include <stdio.h>
2  #include <vector>
3
4  int main()
5  {
6      int n, m;
7      scanf("%d %d", &n, &m);
8      std::vector<int> times;
9      times.resize(n);

```

```

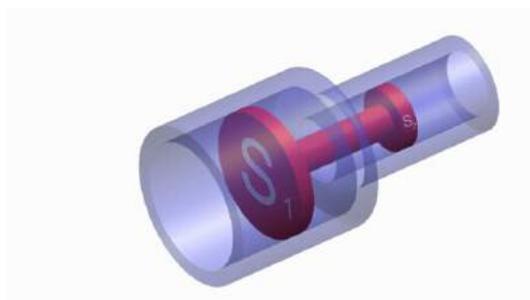
10
11     times[0] = 0;
12     scanf("%d", &times[1]);
13     for (int i = 2; i < n; i++) {
14         scanf("%d", &times[i]);
15         times[i] += times[i - 1];
16     }
17
18     for (int i = 0; i < m; i++) {
19         int a, b;
20         scanf("%d %d", &a, &b);
21         if (a > b)
22             std::swap(a, b);
23         printf("%d\n", times[b - 1] - times[a - 1]);
24     }
25 }

```

3.2. Блок заданий 2

Задача 3.2.1. (15 баллов)

Элемент поршневого соединения в механизме погрузки-разгрузки имеет вид как на рисунке. Площади сечения цилиндрических труб $S_1 = 25 \text{ см}^2$ и $S_2 = 5 \text{ см}^2$. Поршни соединены жёстким стержнем. Вначале давление во всех частях системы равно атмосферному $p_0 = 101 \text{ кПа}$, а объём между поршнями равен $V_0 = 0.5 \text{ м}^3$. На сколько процентов нужно поднять давление в цилиндре меньшего диаметра (правая часть системы), чтобы поршни сместились на 5 см. Давление в левой части системы остаётся прежним. Трения нет, температура в системе постоянна.



Ответ дать в процентах с точностью до сотых.

Решение

После смещения поршня влево, внутренний объём изменится на

$$\Delta V = xS_1 - xS_2$$

Поскольку температура и масса газа между поршнями неизменна, то

$$p_0 V_0 = p_2 (V_0 + \Delta V) = p_2 (V_0 + x(S_1 - S_2)), \quad (2.1)$$

где p_2 – новое давление между поршнями. Из условия равновесия поршня:

$$p_0 S_1 - p_2 S_1 - px S_2 + p_2 S_2 = 0 \quad (2.2)$$

Решая совместно уравнения (2.1, 2.2) относительно px и p_2 получим:

$$p_2 \rightarrow \frac{p_0 V_0}{x S_1 - x S_2 + V_0}, \quad px \rightarrow \frac{p_0 (-x S_1^2 + x S_1 S_2 - S_2 V_0)}{S_2 (-x S_1 + x S_2 - V_0)}$$

$$px \rightarrow 101080.8,$$

$$p_2 \rightarrow 100979.8$$

$$100 \frac{p_0 - px}{p_0} = 0.08\%$$

Ответ: 0.08 %.

Задача 3.2.2. (15 баллов)

Злоумышленник сунулся в ваш умный дом, но сработала сигнализация, и человеку пришлось спешно ретироваться. Ультразвуковой датчик над дверью измеряет скорость удаляющегося субъекта. Частота сигнала датчика 40.00 КГц. Определите скорость субъекта, если отражённый от него сигнал приходит на частоте 39.60 КГц. Скорость звука равна 340 м/с.

Ответ дать в м/с с точностью до десятых.

Решение

Датчик работает на эффекте Доплера. Частота отражённого сигнала v :

$$v = \nu_0 \frac{v - u_{2x}}{v - u_{1x}} \quad (1)$$

Здесь ν_0 – частота исходного сигнала, v – скорость звука, u_{1x} – скорость источника по отношению к среде распространения сигнала, u_{2x} – скорость движения приёмника. Сначала источник покоится, приёмник удаляется. После отражения уже источник (отражённый сигнал) удаляется, приёмник покоится. Применяя эту формулу два раза (1) два раза, найдём частоту сигнала, фиксируемого приёмником ультразвукового датчика:

$$v = \nu_0 \frac{v - u}{v + u}$$

Выразим отсюда скорость u объекта:

$$u = v \frac{\nu_0 - v}{v + \nu_0}$$

И подставим значения частот и скорости звука. Получим с точностью до десятых долей:

$$u = 1.7 \text{ м/с.}$$

Ответ: 1.7 м/с.

3.3. Блок заданий 3

Задача 3.3.1. (10 баллов)

Для обеспечения безопасности все данные между любыми устройствами умного города передаются в зашифрованном виде. Мы не будем вдаваться в подробности используемого метода шифрования, скажем лишь, что для успешного дешифрования сообщения необходимо сформировать минимальное возможное число, удовлетворяющее следующим условиям:

- в десятичной записи числа есть только цифры 3 и 9;
- в десятичной записи числа есть ровно A цифр 3;
- в десятичной записи числа есть ровно B цифр 9;
- в десятичной записи числа есть ровно C вхождений подстроки «39»;
- в десятичной записи числа есть ровно D вхождений подстроки «93».

Ваша задача написать программу, которая по заданным значениям A , B , C и D находит такое число.

Формат входных данных

В единственной строке даны 4 целых числа A , B , C и D ($1 \leq A, B, C, D \leq 10^6$).

Формат выходных данных

В единственной строке выведите минимальное число, удовлетворяющее заданным условиям. Если такого числа не существует, выведите «-1».

Пример №1

Стандартный ввод
3 2 2 1
Стандартный вывод
33939

Пример №2

Стандартный ввод
5 4 4 1
Стандартный вывод
-1

Решение

Для существования описанного числа значения A , B , C , D должны удовлетворять некоторым условиям:

- разность между C и D не должна превышать 1;
- должно быть не меньше, чем C и D ;
- B должно быть не меньше, чем C и D .

Если одно из условий не выполняется, в ответ сразу можно вывести «-1». В противном случае строим минимальное число, удовлетворяющее всем условиям и выводим его.

Пример программы-решения

Ниже представлено решение на языке C++

```

1  #include <cstdio>
2  #include <string>
3  #include <cmath>
4  using namespace std;
5
6  FILE *f = nullptr;
7  FILE *fo = nullptr;
8
9  int sol()
10 {
11     int a1, a2, a3, a4;
12     fscanf(f, "%d%d%d%d", &a1, &a2, &a3, &a4);
13     if (a3 > a1 || a3 > a2 || a4 > a1 || a4 > a2 || a3 - a4 > 1 || a4 - a3 > 1)
14     {
15         fprintf(fo, "-1\n");
16         return 0;
17     }
18     string s;
19     s = '3';
20     --a1;
21     int i = 0;
22     while (a1&&a2&&a3&&a4)
23     {
24         if (s[i] == '3')
25         {
26             s += '9';
27             --a2, --a3, ++i;
28         }
29         if (s[i] == '9')
30         {
31             s += '3';
32             --a1, --a4, ++i;
33         }
34     }
35
36     if (!a1)
37     {
38         if (s[s.length() - 1] == '3')
39         {
40             if ((!a2 && (a3 || a4)) || a3 > 1 || a4 > 1)
41             {
42                 fprintf(fo, "-1\n");
43                 return 0;
44             }
45             if (a2 && a3 == 1)
46             {

```

```

47         s += '9';
48         a3 = 0;
49         --a2;
50     }
51     if (!a2 && a4)
52     {
53         fprintf(fo, "-1\n");
54         return 0;
55     }
56     if (a2 && a4 == 1)
57     {
58         s = "9" + s;
59         --a2;
60         a4 = 0;
61     }
62     for (int i = 0; i < s.length(); ++i)
63     {
64         if (i + 1 == s.length())
65             for (int i = 0; i < a2; ++i)
66                 fprintf(fo, "9");
67         fprintf(fo, "%c", s[i]);
68     }
69     return 0;
70 }
71 }
72
73 if (!a2)
74 {
75     if (s[s.length() - 1] == '3')
76     {
77         if (a3 || a4)
78         {
79             fprintf(fo, "-1\n");
80             return 0;
81         }
82         for (int i = 0; i < a1; ++i)
83             fprintf(fo, "3");
84         fprintf(fo, "%s\n", s.c_str());
85         return 0;
86     }
87
88
89     if (s[s.length() - 1] == '9')
90     {
91         if ((!a1 && (a3 || a4)) || a3 || a4 > 1)
92         {
93             fprintf(fo, "-1\n");
94             return 0;
95         }
96         if (a1 && a4 == 1)
97         {
98             s += '3';
99             --a1;
100            a4 = 0;
101        }
102        for (int i = 0; i < a1; ++i)
103            fprintf(fo, "3");
104        fprintf(fo, "%s\n", s.c_str());
105        return 0;
106    }

```

```
107     }
108
109     if (!a3)
110     {
111         if (!a4)
112         {
113             for (int i = 0; i < a1; ++i)
114                 fprintf(fo, "3");
115             if (s[s.length() - 1] != '3')
116             {
117                 fprintf(fo, "%s\n", s.c_str());
118                 for (int i = 0; i < a2; ++i)
119                     fprintf(fo, "9");
120             } else
121             {
122                 for (int i = 0; i < s.length(); ++i)
123                 {
124                     if (i + 1 == s.length())
125                         for (int j = 0; j < a2; ++j)
126                             fprintf(fo, "9");
127                     fprintf(fo, "%c", s[i]);
128                 }
129             }
130         }
131         return 0;
132     }
133
134     if (s[s.length() - 1] == '3')
135     {
136         if ((a4 && !a2) || a4 > 1)
137         {
138             fprintf(fo, "-1\n");
139             return 0;
140         } else if (a4 == 1 && a2)
141         {
142             //s = "9" + s;
143             --a2;
144             --a4;
145             fprintf(fo, "9");
146         }
147         for (int i = 0; i < a1; ++i)
148             fprintf(fo, "3");
149         for (int i = 0; i < s.length(); ++i)
150         {
151             if (i + 1 == s.length())
152                 for (int j = 0; j < a2; ++j)
153                     fprintf(fo, "9");
154             fprintf(fo, "%c", s[i]);
155         }
156         return 0;
157     }
158
159     }
160
161     if (s[s.length() - 1] == '9')
162     {
163         if ((a4 && !a1) || a4 > 1)
164         {
165             fprintf(fo, "-1\n");
166             return 0;
167         }
168     }
```

```
167         if (a4 == 1 && a1)
168         {
169             s += "3";
170             --a1;
171             --a4;
172         }
173         for (int i = 0; i < a1; ++i)
174             fprintf(fo, "3");
175         fprintf(fo, "%s\n", s.c_str());
176         for (int j = 0; j < a2; ++j)
177             fprintf(fo, "9");
178         return 0;
179     }
180 }
181 }
182
183 if (!a4)
184 {
185     if (!a3)
186     {
187         for (int i = 0; i < a1; ++i)
188             fprintf(fo, "3");
189         fprintf(fo, "%s\n", s.c_str());
190         for (int i = 0; i < a2; ++i)
191             fprintf(fo, "9");
192         return 0;
193     }
194
195     if (s[s.length() - 1] == '3')
196     {
197         if ((a3 && !a2) || a3 > 1)
198         {
199             fprintf(fo, "-1\n");
200             return 0;
201         } else if (a3 == 1 && a2)
202         {
203             s += "9";
204             --a2;
205             --a3;
206         }
207         for (int i = 0; i < a1; ++i)
208             fprintf(fo, "3");
209         fprintf(fo, "%s\n", s.c_str());
210         for (int j = 0; j < a2; ++j)
211             fprintf(fo, "9");
212         return 0;
213     }
214 }
215
216 if (s[s.length() - 1] == '9')
217 {
218     if (a3)
219     {
220         fprintf(fo, "-1\n");
221         return 0;
222     }
223     for (int i = 0; i < a1; ++i)
224         fprintf(fo, "3");
225     fprintf(fo, "%s\n", s.c_str());
226     for (int i = 0; i < a2; ++i)
```

```
227         fprintf(fo, "9");
228     return 0;
229
230     }
231 }
232 }
233
234 int main()
235 {
236     f = stdin;
237     fo = stdout;
238     sol();
239     return 0;
240 }
```

Задача 3.3.2. (10 баллов)

Хороший робот-пылесос должен уметь убираться во всем доме. Пылесос Ёмкий Трудолюбивый Роботизированный (П.Ё.Т.Р.) как раз из таких. Однако он намного умнее своих собратьев: он умеет заряжаться до такого состояния, чтобы его заряда хватило на уборку всей квартиры. Для этого ему необходимо знать площадь, на которой требуется провести уборку. Благо, на этот случай у него есть план квартиры и место нахождения самого П.Ё.Т.Р.а. Помогите П.Ё.Т.Р.у найти площадь на которой он может навести порядок.

Формат входных данных

В первой строке даны два целых числа W и H – размеры квартиры

В следующих H строках содержится W символов, описывающих квартиру:

'#' - стена

'.' - пол

'@' - пылесос П.Ё.Т.Р.

Формат выходных данных

Число клеток, которые П.Ё.Т.Р. может убрать в квартире.

Пылесос в начальной точке всегда стоит на полу, и эта клетка тоже считается необходимой для уборки.

Гарантируется, что пылесос всегда находится внутри квартиры.

Квартира всегда является замкнутым контуром.

Пылесос не умеет ходить по диагонали.

Пример №1

Стандартный ввод
5 5 ##### #...# #.@.# #...# #####
Стандартный вывод
9

Пример №2

Стандартный ввод
8 11#####. #....#. .###.##. #....#. .##.@.#. #....#. .#####.#.#.###.
Стандартный вывод
16

Пример программы-решения

Ниже представлено решение на языке C++

```

1  #include <string>
2  #include <stdio.h>
3  #include <cstdio>
4  #include <iostream>
5  #include <vector>
6  #include <queue>
7
8  int main()
9  {
10     int w;
11     int h;
12     scanf("%d %d", &w, &h);
13
14     std::vector<std::string> arr;
15     arr.resize(h);
16     for (int y = 0; y < h; y++)
17     {
18         std::cin >> arr[y];
19     }
20
21     bool was_added = true;
22     int painted = 1; // petr itself.

```

```

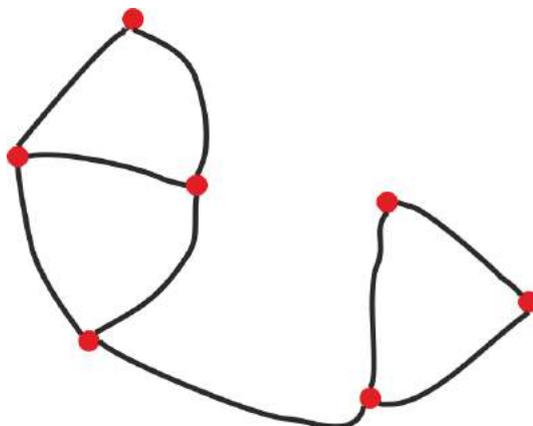
23     while (was_added)
24     {
25         was_added = false;
26
27         for (int y = 1; y < h - 1; y++)
28             for (int x = 1; x < w - 1; x++)
29                 {
30                     if ((arr[y][x] == '.') &&
31                         (arr[y - 1][x] == '@' ||
32                          arr[y + 1][x] == '@' ||
33                          arr[y][x - 1] == '@' ||
34                          arr[y][x + 1] == '@'))
35                         {
36                             arr[y][x] = '@';
37                             was_added = true;
38                             painted++;
39                         }
40                 }
41     }
42
43     printf("%d\n", painted);
44 }

```

Задача 3.3.3. (15 баллов)

При первоначальной разработке топологии сети передачи данных между различными системами умного города архитектор очень сильно болел, вследствие чего сеть была спроектирована не оптимально. Со временем, когда к сети было подключено большое количество управляющих систем, недостатки проектировки стали сильно влиять на скорость передачи данных. Для полной реорганизации сети потребовалось бы огромное количество ресурсов, поэтому администрацией было принято решение исправлять то, что уже есть.

В ходе долгих обсуждений эксперты пришли к выводу, что каждый участок сети должен быть организован таким образом, чтобы в нём содержалось ровно N «треугольных связей». Под «треугольными связями» эксперты подразумевают произвольную тройку узлов, между каждой парой из которых имеется связь. Например, на рисунке таких треугольных связей 3.



В результате некоторых исследований различных вариантов эксплуатации треугольных связей была выявлена закономерность, что когда количество узлов, внутри

которых устанавливается ровно N треугольных связей, минимально, сеть работает с наибольшей пропускной способностью. При этом вся сеть должна быть связной.

Ваша задача по точно заданному числу треугольных связей N определить минимальное количество узлов M , необходимых для обеспечения этого количества связей.

Формат входных данных

В единственной строке дано целое число N ($1 \leq N \leq 10^5$).

Формат выходных данных

Выведите единственное число M .

Пример №1

Стандартный ввод
1
Стандартный вывод
3

Пример №2

Стандартный ввод
3
Стандартный вывод
5

Решение

Минимальное число узлов, которое может обеспечить хотя бы одну треугольную связь, – три. На трёх узлах можно построить ровно одну треугольную связь, соединив каждый узел с остальными двумя. При добавлении четвёртого узла и соединении его с двумя из уже имеющихся, образуется ещё одна треугольная связь (итого 2). Протянув последнюю недостающую связь между узлами, получаем ещё 2 новых треугольных связи (итого 4). Обратим внимание, что ровно три треугольных связи на четырёх узлах построить невозможно, несмотря на то, что можно построить 4 таких связи.

В общем случае при добавлении каждого последующего узла и поочередного соединения вновь добавленного узла с остальными каждая очередное соединение (начиная со второго) будет создавать на одну треугольную связь больше, чем предыдущее (т.е. добавив первые два соединения создаётся одна треугольная связь, три соединения – ещё плюс две связи, четыре соединения – ещё плюс три связи и т.д.).

Таким образом задачу можно решить простым итерированием, постепенным добавлением по одному узлу и по одному ребру до тех пор, пока очередное ребро не создаст слишком много треугольных связей. В этот момент добавляется ещё один узел, для которого тоже начинаем создавать соединения.

Пример программы-решения

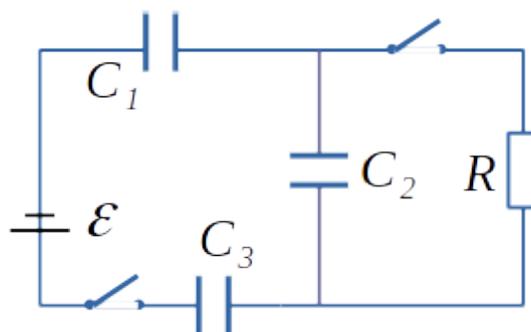
Ниже представлено решение на языке C++

```
1  #include <stdio>
2  #include <string>
3  #include <cmath>
4  using namespace std;
5
6  FILE *f = nullptr;
7  FILE *fo = nullptr;
8
9  int sol()
10 {
11     int k;
12     fscanf(f, "%d", &k);
13
14     int v = 3, c = 1;
15
16     for (;c < k;)
17     {
18         ++v;
19         ++c;
20         for (int i = v - 4, lc = 2; i >= 0 && c < k; --i)
21         {
22             if (c + lc > k)
23                 break;
24             c += lc;
25             ++lc;
26         }
27     }
28
29     fprintf(fo, "%d\n", v);
30     return 0;
31 }
32
33 int main()
34 {
35     f = stdin;
36     fo = stdout;
37     return sol();
38 }
```

3.4. Блок заданий 4

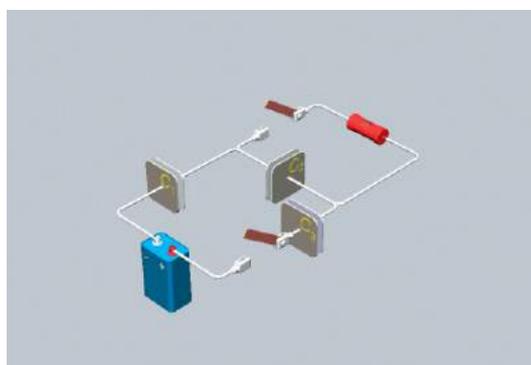
Задача 3.4.1. (15 баллов)

Вы ремонтируете некое устройство в вашем доме. Видите, что оно представляет собой схему, изображенную на рисунке.



Вам необходимо узнать, какой заряд протечёт через резистор R после одновременного замыкания ключей K_1 и K_2 . До замыкания конденсаторы $C_1 = 1\text{мкФ}$, $C_3 = 3\text{мкФ}$ не заряжены, а конденсатор $C_2 = 2\text{мкФ}$ заряжен до разности потенциалов $U_0 = 1\text{В}$. ЭДС батареи 6В .

Ответ дать в микрокулонах с точностью до десятых.



Решение

После замыкания ключей конденсатор C_2 разряжается через R , а C_1 и C_3 заряжаются зарядом q_1 :

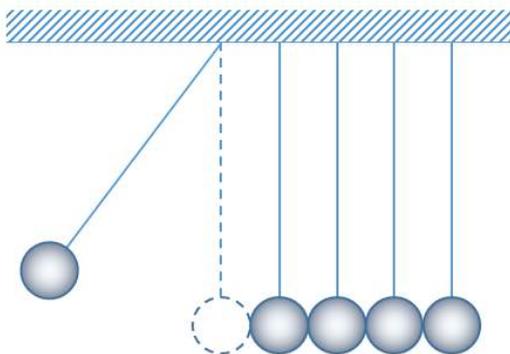
$$\frac{q_1}{C_1} + \frac{q_1}{C_3} = \epsilon \rightarrow q_1 = \frac{\epsilon C_1 C_3}{C_1 + C_3} + C_2 U_0$$

После подстановки численных данных величина заряда, прошедшего через резистор будет равна 6.5 мкКл .

Ответ: 6.5 мкКл .

Задача 3.4.2. (10 баллов)

Для украшения интерьера и снятия стресса вам подарили механическую систему, представляющую собой стальные шарики одинакового радиуса, подвешенные так, что находятся на одной прямой, касаясь друг друга (смотри рисунок).



В инструкции сообщается, что массы шариков разные: начиная с первого, каждый следующий имеет половину массы предыдущего (240 г, 120 г, 60 г, 30 г, 15 г). Вы отклонили первый шарик в сторону так, что нить натянута и образует 30 градусов с вертикалью. Затем вы отпустили шарик с нулевой начальной скоростью.

Определите скорость (в м/с) первого шарика в нижней точке траектории. Удары считать абсолютно упругими. Длина нити 50 см. Ответ представить с точностью до десятых. Ускорение свободного падения считать равным 10 м/с.

Решение

По закону сохранения полной механической энергии для первого шарика:

$$\frac{mv^2}{2} = mgh = mgl(1 - \cos \alpha).$$

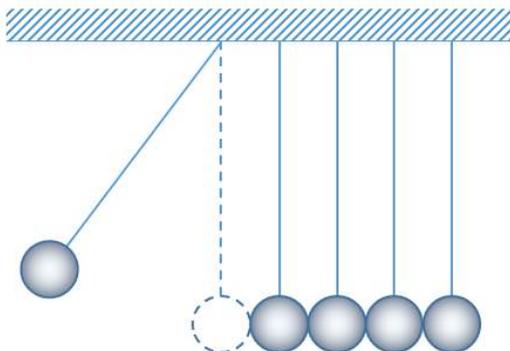
Выразим скорость:

$$v \rightarrow \sqrt{2}\sqrt{gl(1 - \cos \alpha)} = 1.157 \text{ м/с} \approx 1.2 \text{ м/с}.$$

Ответ: 1.2 м/с.

Задача 3.4.3. (5 баллов)

Для украшения интерьера и снятия стресса вам подарили механическую систему, представляющую собой стальные шарики одинакового радиуса, подвешенные так, что находятся на одной прямой, касаясь друг друга (смотри рисунок).



В инструкции сообщается, что массы шариков разные: начиная с первого, каждый следующий имеет половину массы предыдущего (240 г, 120 г, 60 г, 30 г, 15 г). Вы отклонили первый шарик в сторону так, что нить натянута и образует 30 градусов с вертикалью. Затем вы отпустили шарик с нулевой начальной скоростью.

Найдите скорость (в м/с) последнего шарика после удара. Удары считать абсолютно упругими. Длина нити 50 см. Ответы представить с точностью до десятых. Ускорение свободного падения считать равным 10 м/с.

Решение

По закону сохранения полной механической энергии для первого шарика:

$$\frac{mv^2}{2} = mgh = mgl(1 - \cos \alpha).$$

Выразим скорость:

$$v \rightarrow \sqrt{2\sqrt{gl(1 - \cos \alpha)}} = 1.157 \text{ м/с} \approx 1.2 \text{ м/с}.$$

Масса второго шара $m_2 = m_1/2$, n -ого – $m_n = m_1/2^{n-1}$. Из законов сохранения импульса и полной механической энергии для абсолютно упругого удара двух шаров, скорость \vec{u}_n каждого из них после удара равна $\vec{u}_n = 2\vec{v}_c - \vec{v}_n$. Скорость \vec{v}_n – скорость до удара. Поскольку исходно вся цепочка шаров покоилась, то скорость $v_n = 0$ и

$$u_n = 2v_c = 2 \frac{m_{n-1}v_{n-1}}{m_{n-1} + m_n} = 2 \frac{v_{n-1}}{1 + m_n/m_{n-1}} = \frac{2}{1 + 1/2} v_{n-1} = \frac{4}{3} v_{n-1}.$$

Таким образом скорость второго шарика после взаимодействия с первым $u_2 = \frac{4}{3}v_1$, где $v_1 = 1.157$ м/с. Скорость третьего шарика после взаимодействия со вторым

$$u_3 = \frac{4}{3}u_2 = \left(\frac{4}{3}\right)^2 v_1,$$

для n -ого шарика

$$u_n = \left(\frac{4}{3}\right)^{n-1} v_1.$$

Искомая скорость пятого шарика

$$u_5 = \left(\frac{4}{3}\right)^4 v_1 = 3.658 \text{ м/с} \approx 3.7 \text{ м/с}.$$

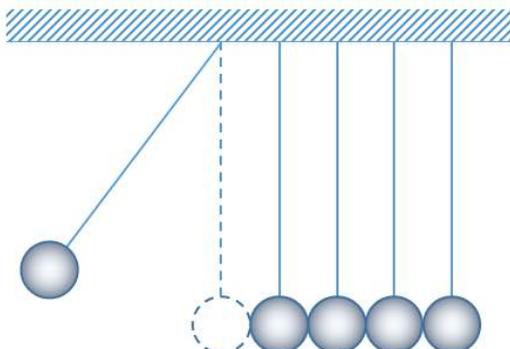
Ответ 3.8 тоже считать верным, но давать меньше баллов. Дело в том, что если здесь подставить округлённое число скорости первого шарика 1.2, то ответ для скорости пятого будет отличаться на одну десятую.

То есть ошибка – если округляют до вычислений.

Ответ: 3.7 м/с.

Задача 3.4.4. (5 баллов)

Для украшения интерьера и снятия стресса вам подарили механическую систему, представляющую собой стальные шарики одинакового радиуса, подвешенные так, что находятся на одной прямой, касаясь друг друга (смотри рисунок).



В инструкции сообщается, что массы шариков разные: начиная с первого, каждый следующий имеет половину массы предыдущего (240г, 120г, 60г, 30г, 15г). Вы отклонили первый шарик в сторону так, что нить натянута и образует 30 градусов с вертикалью. Затем вы отпустили шарик с нулевой начальной скоростью.

Найдите кинетическую энергию (в джоулях) последнего шарика после удара. Удары считать абсолютно упругими. Длина нити 50см. Ответы представить с точностью до десятых. Ускорение свободного падения считать равным 10 м/с.

Решение

По закону сохранения полной механической энергии для первого шарика:

$$\frac{mv^2}{2} = mgh = mgl(1 - \cos \alpha).$$

Выразим скорость:

$$v \rightarrow \sqrt{2\sqrt{gl(1 - \cos \alpha)}} = 1.157 \text{ м/с} \approx 1.2 \text{ м/с}.$$

Масса второго шара $m_2 = m_1/2$, n -ого — $m_n = m_1/2^{n-1}$. Из законов сохранения импульса и полной механической энергии для абсолютно упругого удара двух шаров, скорость \vec{u}_n каждого из них после удара равна $\vec{u}_n = 2\vec{v}_c - \vec{v}_n$. Скорость \vec{v}_n — скорость до удара. Поскольку исходно вся цепочка шаров покоилась, то скорость $v_n = 0$ и

$$u_n = 2v_c = 2\frac{m_{n-1}v_{n-1}}{m_{n-1} + m_n} = 2\frac{v_{n-1}}{1 + m_n/m_{n-1}} = \frac{2}{1 + 1/2}v_{n-1} = \frac{4}{3}v_{n-1}.$$

Таким образом скорость второго шарика после взаимодействия с первым $u_2 = \frac{4}{3}v_1$, где $v_1 = 1.157$ м/с. Скорость третьего шарика после взаимодействия со вторым

$$u_3 = \frac{4}{3}u_2 = \left(\frac{4}{3}\right)^2 v_1,$$

для n -ого шарика

$$u_n = \left(\frac{4}{3}\right)^{n-1} v_1.$$

Искомая скорость пятого шарика

$$u_5 = \left(\frac{4}{3}\right)^4 v_1 = 3.658 \text{ м/с} \approx 3.7 \text{ м/с}.$$

Ответ 3.8 тоже считать верным, но давать меньше баллов. Дело в том, что если здесь подставить округлённое число скорости первого шарика 1.2, то ответ для скорости пятого будет отличаться на одну десятую.

То есть ошибка – если округляют до вычислений.

Кинетическая энергия n -ого шарика

$$E_n = \frac{m_n u_n^2}{2} = \frac{1}{2} \frac{m_1}{2^{n-1}} \left(\frac{4}{3}\right)^{2(n-1)} v_1^2.$$

После упрощений

$$E_n = \left(\frac{8}{9}\right)^{n-1} \frac{m_1 v_1^2}{2}$$

Искомая энергия пятого шарика

$$E_5 = \left(\frac{8}{9}\right)^{5-1} \frac{m_1 v_1^2}{2} = \left(\frac{8}{9}\right)^4 \frac{1}{2} 0.24 \cdot 1.157^2 = 0.1 \text{ Дж}.$$

Ответ: 0.1 Дж.

3.5. Блок заданий 5

Задача 3.5.1. (20 баллов)

Егор – программист, настраивающий систему освещения умным городом. Иногда Егору бывает скучно на работе, и он придумывает себе различные алгоритмические задачки. Однажды Егор выписал на лист бумаги N строчек, каждая из которых состояла из M символов «а» и «b». Символы чередовались в произвольном порядке. Егор заметил, что в полученной табличке не было ни одного столбца, полностью состоящего из букв «а». Ему стало интересно, какое минимальное суммарное количество циклических сдвигов строк потребуется, чтобы в таблице появился хотя бы один такой столбец. Каждая строка сдвигается независимо от всех остальных.

Под циклическим сдвигом понимается смещение всех символов строки на одну позицию вправо или влево, при этом при сдвиге вправо последний символ исходной строки переносится на место первого, а при сдвиге влево – первый на место последнего.

Сдвиг вправо «bbbaba» → «abbbab».

Сдвиг влево «bbbaba» → «bbabab».

Формат входных данных

В первой строке даны два числа N и M ($1 \leq N \leq 100$; $1 \leq M \leq 10^4$). Следующие N строк содержат строки длиной M состоящие из символов «a» и «b».

Формат выходных данных

Выведите единственное число K – минимальное число циклических сдвигов, при котором хотя бы один из столбцов таблицы будет состоять только из символов «a». Если такое состояние получить невозможно, выведите «-1».

Пример №1

Стандартный ввод
3 6
ababab
bbbabb
babbbb
Стандартный вывод
2

Пример №2

Стандартный ввод
3 3
aba
bbb
bba
Стандартный вывод
-1

Решение

Если во входных данных есть хотя бы одна строка, состоящая только из букв «b», то решения задача не имеет и в ответ выводится «-1». В противном случае ответ существует всегда.

Для каждого символа каждой строки посчитаем расстояние до ближайшего слева или справа символа «a». Для каждого столбика посчитаем сумму этих значений. Ответом является минимальная из этих сумм.

Пример программы-решения

Ниже представлено решение на языке C++

```

1  #include <stdio.h>
2  #include <vector>
3
4  int main()

```

```

5 {
6     int n, m;
7     scanf("%d %d", &n, &m);
8     std::vector<int> times;
9     times.resize(n);
10
11     times[0] = 0;
12     scanf("%d", &times[1]);
13     for (int i = 2; i < n; i++) {
14         scanf("%d", &times[i]);
15         times[i] += times[i - 1];
16     }
17
18     for (int i = 0; i < m; i++) {
19         int a, b;
20         scanf("%d %d", &a, &b);
21         if (a > b)
22             std::swap(a, b);
23         printf("%d\n", times[b - 1] - times[a - 1]);
24     }
25 }

```

Задача 3.5.2. (30 баллов)

Данная задача является исследовательской, поэтому ограничения на количество попыток нет.

Мы взяли картинку размером 1920×1080 , некоторым алгоритмом уменьшили её в два раза по ширине и в два раза по высоте, получили картинку размером 960×540 .

Ваша задача растянуть картинку 960×540 обратно до 1920×1080 таким образом, чтобы разница между оригинальной картинкой и вашей растянутой была минималь-

ной. Под разницей будем понимать средний квадрат разности $D = \frac{\sum_{i=1}^{w \times h} (p_{org}[i] - p_{rec}[i])^2}{w \times h}$,

где $p_{org}[i]$ - значения пикселей оригинальной картинки, $p_{rec}[i]$ - значения пикселей восстановленной вами картинки, w и h - ширина и высота изображения (1920 и 1080 в данном случае).

Формат входных данных

В первой строке выведите два числа w и h - ширину и высоту полученного вами изображения (1920 и 1080). В следующей строке выведите закодированное в Base64 изображение в градациях серого.

Формат выходных данных

Аналогично формату входных данных. В первой строке выведите два числа w и h - ширину и высоту полученного вами изображения (1920 и 1080). В следующей строке выведите закодированное в Base64 (<https://ru.wikipedia.org/wiki/Base64>) изображение в градациях серого (https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D1%82%D0%B5%D0%BD%D0%BA%D0%B8_%D1%81%D0%B5%D1%80%D0%BE%D0%B3%D0%BE).

Примечание: В качестве примера входных и выходных данных предлагаем вам скачать файлы. `sample_input.txt` - входной файл

(https://drive.google.com/file/d/1dloLs5pcLKvVybrCFEG5y6E_ahI5_J_J/view),
sample_output.txt - выходной

(https://drive.google.com/file/d/107sLUq9s_kAqQzFuz6W_JK00FCjwRqcL/view)

Обращаем ваше внимание, что в примере размер большой картинки 720×480 , а уменьшенной 360×240 соответственно. Кроме того, что sample_output.txt является корректным примером выхода вашей программы, он так же является оригиналом картинки уменьшенной до 360×240 тем же алгоритмом, что и изображение из задания.

Входное изображение во всех тестах одинаковое. Каждый тест засчитывается, если величина D , подсчитанная на основании разности выхода вашей программы и оригинального изображения не превосходит некоторую величину погрешности ϵ . Тесты упорядочены в порядке убывания ϵ . Величина ϵ в первом тесте соответствует величине D для метода ближайшего соседа.

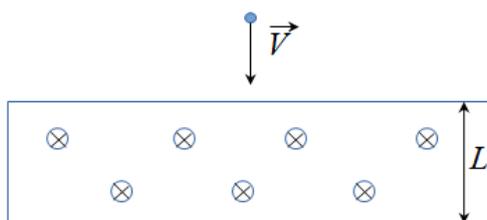
Решение

Задача является исследовательской и идеального решения не имеет. Для получения большего количества баллов за задачу необходимо реализовать хороший метод двумерной интерполяции.

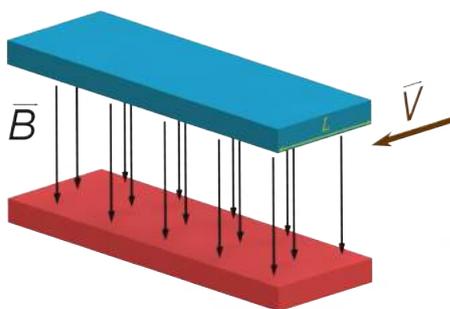
3.6. Блок заданий 6

Задача 3.6.1. (10 баллов)

Электрон влетает в область пространства с однородным магнитным полем (смотри рисунок) перпендикулярно линиям индукции магнитного поля.



Индукция магнитного поля равна 10^{-3} Тл, скорость электрона $V = 3 \cdot 10^6$ м/с. Расстояние $L = 10$ см. Определите угол (в градусах) между векторами скорости влёта и вылета из области пространства с магнитным полем если начальная скорость была перпендикулярна границе этой области.



Решение

На каждый движущийся заряд со стороны магнитного поля действует сила Лоренца:

$$F = q \cdot V \cdot B \cdot \sin(\vec{V} \vec{B})$$

Радиус найдем из уравнения Ньютона:

$$F = ma$$

$F = qVB$ - сила Лоренца, a - центростремительное ускорение

$$qVB = m \frac{V^2}{R}$$

$$R = \frac{mV}{aB} = 0.0171 \text{ м}$$

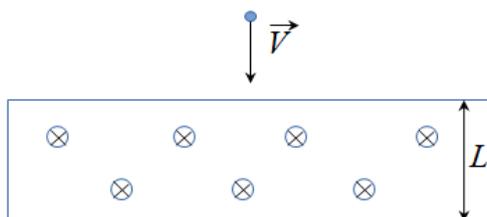
это меньше чем ширина области магнитного поля, следовательно электрон не выйдет за пределы поля с противоположной стороны от точки влёта, будет двигаться по полуокружности в м.п. и выйдет в обратную сторону, также под углом 90 градусов к границе области м.п.

Следовательно, угол между скоростями влёта и вылета: $\alpha = 180^\circ$.

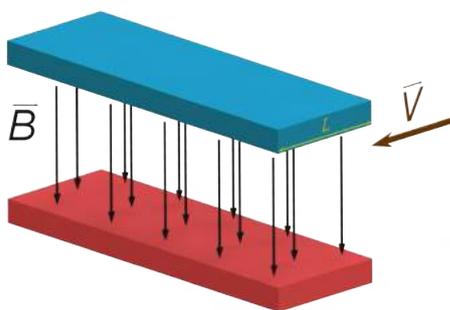
Ответ: 180° .

Задача 3.6.2. (10 баллов)

Электрон влетает в область пространства с однородным магнитным полем (смотри рисунок) перпендикулярно линиям индукции магнитного поля.



Индукция магнитного поля равна 10^{-3} Тл, скорость электрона $V = 3 \cdot 10^6$ м/с. Расстояние $L = 10$ см. Найдите величину изменение импульса электрона за время нахождения в магнитном поле (ответ представить в кг·м/с на 10^{-24} , округлить до десятых).



Решение

Изменение импульса электрона:

$$\Delta \vec{p} = \vec{p}_2 - \vec{p}_1$$

В проекциях на направление оси:

$$\Delta p = -p_2 - p_1$$

Сила Лоренца перпендикулярна вектору скорости и не меняет его модуль, поэтому:

$$p_2 = p_1$$

$$|\Delta p| = 2p = 2mV$$

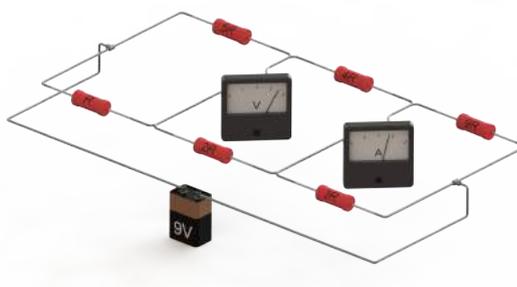
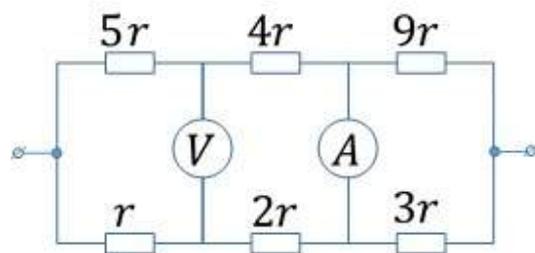
$$\Delta p = 2mV$$

$$\Delta p = 5.466 \cdot 10^{-24} \text{ кг} \cdot \text{м/с на } 10^{-24}$$

Ответ: 5.5 кг · м/с на 10^{-24}

Задача 3.6.3. (10 баллов)

Электрическая схема приведена на рисунке. Приборы считать идеальными. Определите показания амперметра, если на выводы цепи подано напряжение 9 В, а $r = 90$ Ом.



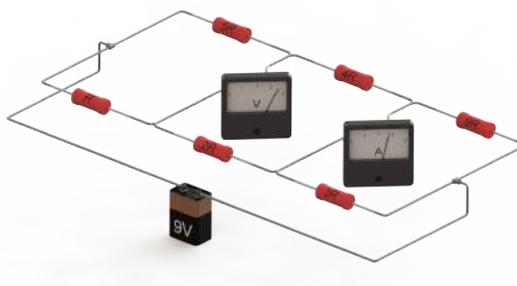
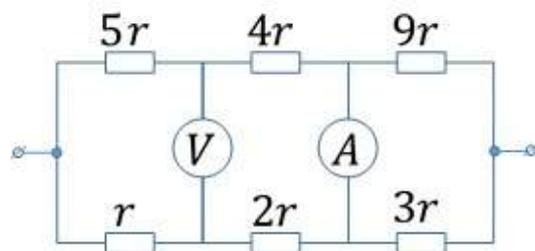
Решение

Вольтметр идеальный. Следовательно, ток через вольтметр равен нулю. Амперметр идеальный – является перемычкой в сбалансированном мосте. Значит через амперметр ток не течёт, показания амперметра $I_A = 0$.

Ответ: 0.

Задача 3.6.4. (10 баллов)

Электрическая схема приведена на рисунке. Приборы считать идеальными. Определите показания вольтметра, если на выводы цепи подано напряжение 9 В, а $r = 90$ Ом.



Решение

Вольтметр идеальный. Следовательно, ток через вольтметр равен нулю. Амперметр идеальный – является перемычкой в сбалансированном мосте. Значит через амперметр ток не течёт, показания амперметра $I_A = 0$.

Сопротивление верхней цепи $18r$, нижней $6r$. Отличаются в три раза, следовательно, и токи в верхней и нижней ветвях отличаются в три раза.

Общее сопротивление равно

$$R = \frac{18r \cdot 6r}{18r + 6r} = \frac{9}{2}r.$$

Полный ток по закону Ома:

$$I = \frac{U}{R} = \frac{2U}{9r} = \frac{3U}{18r} + \frac{1U}{18r}.$$

Падения напряжения на резисторе $5r$ равно (ток равен одной четвёртой от общего):

$$U_5 = \frac{1}{4}I \cdot 5r = \frac{U}{18r}5r = \frac{5U}{18} = 2,5 \text{ В.}$$

Падения напряжения на резисторе r равно (ток равен трём четвёртым от общего):

$$U_r = \frac{3}{4}I \cdot r = \frac{3U}{18} = 1,5 \text{ В.}$$

Значит показание вольтметра

$$U_V = U_5 - U_r = 1 \text{ В.}$$

Ответ: 1.