

Методические указания по Отраслевой олимпиаде школьников «Газпром», профиль «Информационные и коммуникационные технологии»

Учебное пособие для подготовки к олимпиаде

Под редакцией Ю.Б.Буркатовской, В.В.Видмана

Тематика заданий

1. Логическая цепочка
2. Алгоритмы на графах
3. Логика
4. Обработка большого массива данных
5. Графы
6. Синтез дискретных устройств
7. Комбинаторика
8. Кодирование и декодирование
9. Логика
10. Кодирование и декодирование

1. Синтез дискретных устройств

Задача 1. Работа Серсеи заключается в ведении некоторой документации о текущем состоянии нефтеносного слоя на Северном месторождении. Ей нужно было собрать данные с 4 серверов. Сначала она подключилась к первому серверу с помощью пароля (13 2 171) и зафиксировала данные с него. Затем она сделала тоже самое с серверами 2 и 3 используя пароли (25 2 671) и (3 4 121). А вот с сервером 4 возникла проблема, она смогла вспомнить только часть пароля (5 4 ?), она никак не могла вспомнить последнего числа. Если сбрасывать пароль, то придется писать объяснительную. Может всё-таки попытаться вспомнить пароль? Одно она точно помнит, во всех паролях есть какая-то связь.

В качестве ответа запишите недостающее число.

Решение задачи 1.

Для начала нужно найти взаимосвязь между числами. Так как во всех числах встречаются только цифры от 0 до 7, то можно предположить, что числа представлены в восьмеричной системе счисления. Переведем их в десятичную.

$$13\ 2\ 171 = 11\ 2\ 121$$

$$25\ 2\ 671 = 21\ 2\ 441$$

$$3\ 4\ 121 = 3\ 4\ 81$$

Если посмотреть на эти числа можно увидеть, что третье число – это первое число в степени равной второму числу, т. е. $11^2=121$; $21^2=441$; $3^4=81$.

Переведем четвертый пароль в десятичную систему счисления и возведем первое число в степень: $5^4=625$. Число 625 нужно перевести обратно в восьмеричную систему счисления. $625_{(10)}=1161_{(8)}$.

Ответ: 5 4 1161

2. Алгоритмы на графах

Задача 2.1. Геологи Иннокентий и Серсея написали годовой отчет о своей работе. Теперь его нужно подписать у N руководителей компании Дотракия. Процедура подписи не совсем прозрачна, и у геологов имеются лишь отрывочные сведения, кто после кого подписывает документ. Проблема в том, что, если не соблюсти порядок подписей, можно нарваться на неприятности. Например, если Тирион подписывает документы после Дейнерис, и документ попадает к Тириону без подписи Дейнерис, то Тирион в ярости рвет документ, и все нужно начинать сначала. Гарантируется, что правильный порядок подачи документов на подпись существует, но его единственность не обязательна. Помогите геологам получить любой правильный порядок подачи документов на подпись.

Входные данные

В первой строке вводятся число N (натуральное, не превышает 50), и число k (натуральное, не превышает 100). Затем идет k строк, каждая из которых содержит два различных числа в диапазоне от 1 до N , которая определяет порядок подписи. Строка вида “ $p\ q$ ” означает, что начальник p подписывает документ ДО начальника q .

Выходные данные

Выведите N различных чисел от 1 до N , определяющих верный порядок подписи документа.

Пример.

Входные данные

6 7

1 3

3 2

3 6

4 1

4 5

5 6

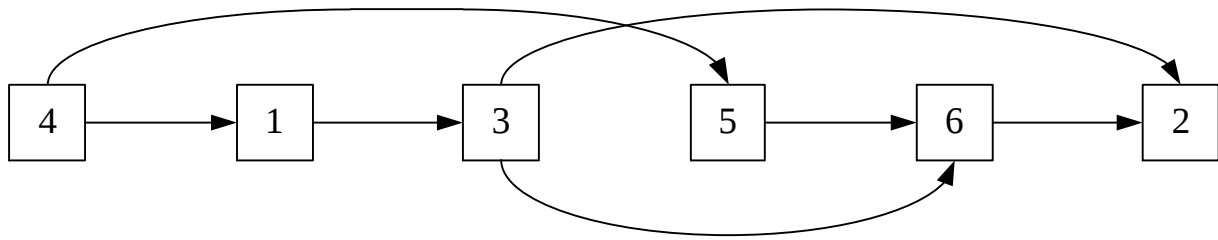
6 2

Выходные данные

4 1 3 5 6 2

Примечание. Возможно данный порядок не единственный, но он удовлетворяет всем условиям.

Решение задачи 2.1. Данную задачу можно представить в виде графа, в котором вершины сопоставлены начальникам, и имеется ориентированное ребро pq , если начальник p подписывает документ раньше, чем начальник q . Тогда задача сводится к **топологической сортировке** графа, то есть к упорядочиванию его вершин таким образом, чтобы все ребра были ориентированы «слева направо».



Для топологической сортировки можно использовать обход графа в глубину. Подробнее о топологической сортировке и использовании для нее обхода в глубину (dfs) можно прочитать, например, здесь:

- https://e-maxx.ru/algo/topological_sort
- <https://informatics.msk.ru/course/view.php?id=6>
- <https://habr.com/ru/post/100953/>

Еще одна идея топологической сортировки состоит в следующем. Находим вершину, в которую не заходит ни одной дуги, помещаем ее в конец списка и удаляем ее из графа. Если граф ациклический, то вершина без заходящих ребер существует всегда. Повторяем до тех пор, пока в списке не окажутся все вершины. Подобный подход подробно описан здесь.

- <https://habr.com/ru/post/66766/>

Также некоторыми участниками олимпиады был реализован алгоритм перестановок: либо находятся все перестановки вершин графа и каждая проверяется на соответствие условиям, либо вершины упорядочиваются в естественном порядке и происходит циклическая проверка условий. Если какое-то условие не выполняется (ребро направлено «справа налево»), вершины переставляются. Подобные алгоритмы, безусловно, являются весьма трудоемкими, но в условиях ограничений на время и память не вводилось, так что подобные решения оценивались положительно.

Приведем некоторые удачные решения участников.

Программа 1. Использует обход в глубину (Python).

```

N,k=map(int,input().split())
a=[[ ]for i in range(N)]
for i in range(k):
    p,q=map(int,input().split())
    p-=1
    q-=1
    a[q].append(p)
used=[True]*N
def dfs(v):
    if not used[v]:
        return True
    for i in a[v]:
        dfs(i)

```

```

    print(v+1,' ')
    used[v]=False
    return True
for i in range(N):
    dfs(i)

```

Программа 2. Использует обход в глубину (C++).

```

#include<vector>
#include<iostream>

using namespace std;
int n,k;
vector<vector<int>>> g;
vector<char> marks;
vector<int>ans;
void dfs(int v) {
    marks[v]=1;
    for (int n:g[v]) if(!marks[n]) dfs(n);
    ans.push_back(v);
}

int main() {
    cin>>n>>k;
    g.resize(n);
    marks.assign(n,0);
    while(k--){
        int p,q;
        cin>>p>>q;
        g[q-1].push_back(p-1);
    }
    for(int v=0; v<n;v++)
        if(!marks[v]) dfs(v);
    for(int v:ans) cout << v+1<< ' ';
}

```

Программа 3. Реализует идею перестановок (Python)

```
n,k=map(int,input().split())
a=[]
b=[]
now=[i+1 for i in range(n)]
prev=[]
for i in range(k):
    x,y=map(int,input().split())
    a.append(x)
    b.append(y)
while now!=prev:
    prev=now.copy()
    for i in range(k):
        if now.index(a[i])>now.index(b[i]):
            now[now.index(a[i])]=b[i]
            now[now.index(b[i])]=a[i]
    print(now)
```

Программа 4. Использует обход в глубину, начинает с вершин без заходящих ребер (C++)

```
#include <iostream>
#include<vector>

using namespace std;

vector<vector<int>> vnext,vprev;
vector<bool> used;

void dfs(int v) {
    for(int u:vprev[v]) if(!used[u]) return;
    cout<< v+1 << " ";
    used[v]=true;
    for(int u:vnext[v]) if(!used[u]) dfs(u);
}
```

```

int main() {
    int n,k;
    cin>>n>>k;
    vnext.resize(n);
    vprev.resize(n);
    for(int i=0;i<k;i++) {
        int p,q;
        cin >>p >>q;
        --p;--q;
        vprev[q].push_back(p);
        vnext[p].push_back(q);
    }
    used.assign(n,0);
    for(int i=0;i<n;++i){
        if(vprev[i].empty()) dfs(i);
    }
}

```

Задача 2.2. Компания Дотракия набирает сотрудников на две вновь открывающихся скважины. Часть сотрудников знакомы между собой, и по итогам этого знакомства некоторые терпеть не могут друг друга. Чтобы не создавать на работе нервную обстановку, руководство не хочет, чтобы такие люди работали вместе. Гарантируется, что это возможно. Помогите руководству решить, какие сотрудники должны работать на первой скважине, а какие – на второй. Известно, что сотрудник с номером 1 всегда работает на первой скважине.

Входные данные

В первой строке вводится натуральное число N , не более 100 – общее количество сотрудников, и число k , не более 200. Затем вводится k строк, по два различных числа в каждой строке, x и y , в диапазоне от 1 до N ; это номера сотрудников, которые терпеть не могут друг друга.

Выходные данные

Выведите список сотрудников первой скважины в произвольном порядке через пробел. Если возможно несколько решений, выведите любое из них.

Пример.

Входные данные

6 5

1 2

1 3

2 5

3 6

4 6

Выходные данные

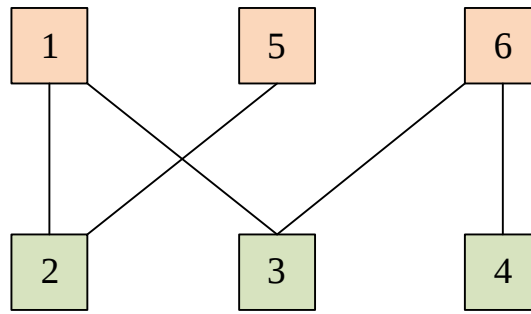
1 5 6

Примечание. Данное решение, возможно, не единственное, но оно удовлетворяет всем условиям.

Решение задачи 2.2. Условия данной задачи можно представить в виде графа, где сотрудникам соответствуют вершины, и они соединяются ребром, если сотрудники ненавидят друг друга. Задача ставится следующим образом: разбить множество вершин на два подмножества таким образом, чтобы любое ребро соединяло вершины из различных подмножеств.

Двудольный граф это граф, вершины которого можно разбить на два множества так, что каждое ребро соединяет вершины из разных множеств. Таким образом, нам дан двудольный граф, требуется найти доли этого графа.

Для данного примера граф выглядит так (доли окрашены в разные цвета).



Задача может быть решена с помощью любого обхода графа, в глубину или в ширину. Вершина, с которой начинаем обход, помечается меткой 1. Если мы переходим в еще не посещенную вершину из вершины с меткой 1, помечаем ее меткой 2, если переходим в вершину из вершины с меткой 2, помечаем ее меткой 1. Если в результате обхода посещены не все вершины, запускаем снова обход с не посещенной вершины.

Подробнее о задаче проверки графа на двудольность и разбиении на доли можно прочитать [здесь](#).

- https://e-maxx.ru/algo/bipartite_checking
- <https://brestprog.by/topics/bipartite/>

Также подходит и такая идея: помещаем вершину 1 в первую долю. Далее просматриваем вершины каждой доли, и, если есть вершина, несовместимая с ней, помещаем ее в другую долю. Зацикливаем этот процесс и повторяем его до тех пор, пока новые вершины не перестают добавляться в доли. Если по долям расписаны не все вершины, одну из этих вершин добавляем в любую долю и повторяем процесс. Эта идея реализована, например, в программе 1.

Приведем некоторые удачные решения участников.

Программа 1. (Python)

```

n,k=map(int,input().split())
mass=[]
mass1=[]
mass2=[]
com=1
mass1.append(int(1))
for i in range(k):
    x,y=map(int,input().split())
    mass.append([x,y])
for i in range(2,n+1):
    for j in range(k):
        if com==1:
            for p in mass1:

```

```

        if mass[j]==[p,i] or mass[j]==[i,p]:
            com=2
    else:
        for p in mass2:
            if mass[j]==[p,i] or mass[j]==[i,p]:
                com=1
    if com==1:
        mass1.append(i)
    else:
        mass2.append(i)
print(mass1,mass2)

```

Программа 2. Использует обход в глубину (C++)

```

#include <iostream>
#include<vector>
#include<algorithm>
using namespace std;

void dfs(int curr,vector<vector<int>> &g,vector<int>&color,vector<int>& vprev){
    if(vprev[curr]==-1) color[curr]=1;
    else color[curr]=3-color[vprev[curr]];
    for(int i=0;i<g[curr].size();++i)
        if(color[g[curr][i]]==0){
            vprev[g[curr][i]]=curr;
            dfs(g[curr][i],g,color,vprev);
        }
}

int main() {
    int k,n;
    cin>>n>>k;
    vector<vector<int>> g(n+1);
    vector<int> color(n+1,0);
    vector<int> vprev(n+1,-1);
    for(int i=0;i<k;i++){
        int x,y;

```

```

    cin>>x>>y;
    x--;y--;
    g[x].push_back(y);
    g[y].push_back(x);
}
for(int i=0;i<n;i++)
    if(color[i]==0) dfs(i,g,color,vprev);
for(int i=0;i<n;i++)
    if(color[i]==1) cout<< i+1 << " ";
}

```

Программа 3. Использует обход в глубину (C++)

```

#include <iostream>
#include<vector>

using namespace std;

vector<int> p;
int color[101],n,k,a[101][101],x,y;

void dfs(int v, int lvl) {
    color[v]=1;
    int i;
    for(i=1;i<=n;i++){
        if ((a[v][i]==1)&&(color[i]!=1)){
            if(lvl) p.push_back(i);
            dfs(i,1-lvl);
        }
    }
}

int main() {
    cin>>n>>k;
    for(int i=0;i<k;i++) {
        cin>>x>>y;
        a[x][y]=a[y][x]=1;
    }
}

```

```
for(int i=1;i<=n;i++) {  
    if(color[i]!=1){  
        p.push_back(i);  
        dfs(i,0);  
    }  
}  
for(int i=0;i<p.size();i++) cout <<p[i] << ' ';
```

3. Логика

Задача 3. В ходе развертывания новой базы для экспедиции группа специалистов отправилась на выбранное место, чтобы собрать необходимую информацию и подготовить лагерь. Команда из пяти человек живет в пяти палатках, установленных в ряд вдоль края поляны под номерами от 1 до 5 (слева направо), работает согласно штатному расписанию, а кроме этого дежурит по кухне, имеет хобби и смотрит любимые сериалы.

- ✓ Известно, что Серсея живет в красной палатке.
- ✓ Дейенерис смотрит сериал «Как я встретил вашу маму».
- ✓ Палатка Дейенерис стоит сразу справа от палатки Петровича.
- ✓ Предпочитающий есть на завтрак манную кашу любит отгадывать кроссворды.
- ✓ Белая палатка имеет номер 1.
- ✓ Сосед того, кто ест мюсли на завтрак, любит играть на гитаре.
- ✓ Сосед того, кто ест яичницу на завтрак, любит читать.
- ✓ Тот, кто любит на завтрак суп, смотрит сериал «Доктор Кто».
- ✓ Зеленая палатка стоит рядом с той, в которой смотрят сериал «Очень странные дела».
- ✓ Джон терпеть не может суп на завтрак.
- ✓ Серсея живет между зеленой и синей палатками.
- ✓ На гитаре играют по соседству с зеленой палаткой.
- ✓ Житель синей палатки занимается йогой.
- ✓ Тот, кто любит компьютерные игры, живет по соседству с увлекающимся йогой.
- ✓ В центральной палатке (под номером 3) смотрят сериал «Друзья».
- ✓ Тирион ест на завтрак яичницу.
- ✓ Обитатель палатки зеленого цвета смотрит сериал «Игра престолов».
- ✓ Живущий в желтой палатке ест на завтрак бутерброды с колбасой.

Напишите все, что вам удалось узнать про Тириона.

Решение задачи 3. Задачу можно решать с помощью таблиц. Построим сначала пустую таблицу с номерами палаток, прочитав утверждения, заполним те ячейки, информация о которых указана прямо в утверждениях.

- ✓ Белая палатка имеет номер 1.
- ✓ В центральной палатке (под номером 3) смотрят сериал «Друзья».

Номер	1	2	3	4	5
Имя					
Цвет палатки	белый				

Любимый сериал			Друзья		
Любимый завтрак					
Хобби					

Далее сопоставляем утверждения:

- ✓ Известно, что Серсея живет в красной палатке.
- ✓ Серсея живет между зеленой и синей палатками.
- ✓ Дейенерис смотрит сериал «Как я встретил вашу маму».
- ✓ Палатка Дейенерис стоит сразу справа от палатки Петровича.

Отсюда:

- ✓ Серсея живет в третьей или четвертой палатке.
- ✓ Дейнерис живет во второй, четвертой или пятой палатке.

Предположим, что Серсея живет в третьей палатке, зеленая палатка стоит слева от нее, синяя – справа. Если в ходе решения придем к противоречию, попробуем другое размещение. Используем утверждения:

- ✓ Зеленая палатка стоит рядом с той, в которой смотрят сериал «Очень странные дела».
- ✓ Житель синей палатки занимается йогой.
- ✓ Обитатель палатки зеленого цвета смотрит сериал «Игра престолов».
- ✓ Живущий в желтой палатке ест на завтрак бутерброды с колбасой.

Номер	1	2	3	4	5
Имя			Серсея		
Цвет палатки	белый	зеленый	красный	синий	желтый
Любимый сериал	Очень странные дела	Игра престолов	Друзья		
Любимый завтрак					бутерброды с колбасой
Хобби				йога	

Вернемся теперь к утверждениям

- ✓ Дейенерис смотрит сериал «Как я встретил вашу маму».
- ✓ Палатка Дейенерис стоит сразу справа от палатки Петровича.

Номер	1	2	3	4	5
Имя			Серсея	Петрович	Дейнерис
Цвет палатки	белый	зеленый	красный	синий	желтый
Любимый сериал	Очень странные дела	Игра престолов	Друзья		Как я встретил вашу маму
Любимый завтрак					бутерброды с колбасой
Хобби				йога	

Значит, Петрович смотрит сериал «Доктор Кто». Учтем теперь утверждения:

- ✓ Тот, кто любит на завтрак суп, смотрит сериал «Доктор Кто».

Номер	1	2	3	4	5
Имя			Серсея	Петрович	Дейнерис
Цвет палатки	белый	зеленый	красный	синий	желтый
Любимый сериал	Очень странные дела	Игра престолов	Друзья	Доктор Кто	Как я встретил вашу маму
Любимый завтрак				суп	бутерброды с колбасой
Хобби				йога	

Мы не использовали утверждения

- ✓ Предпочитающий есть на завтрак манную кашу любит отгадывать кроссворды.
- ✓ Сосед того, кто ест мюсли на завтрак, любит играть на гитаре.
- ✓ Сосед того, кто ест яичницу на завтрак, любит читать.
- ✓ Джон терпеть не может суп на завтрак.
- ✓ На гитаре играют по соседству с зеленой палаткой.

- ✓ Тот, кто любит компьютерные игры, живет по соседству с увлекающимся йогой.
- ✓ Тирион ест на завтрак яичницу.

Можно сделать следующие выводы:

- ✓ На гитаре играют в первой или третьей палатке.
- ✓ Серсея не ест на завтрак мюсли (иначе на гитаре играли бы в зеленой палатке).
- ✓ Серсея также не ест на завтрак яичницу, значит, ей остается манная каша.
- ✓ Серсея увлекается кроссвордами, значит, на гитаре играют в первой палатке.
- ✓ Дейнерис увлекается компьютерными играми.

Номер	1	2	3	4	5
Имя			Серсея	Петрович	Дейнерис
Цвет палатки	белый	зеленый	красный	синий	желтый
Любимый сериал	Очень странные дела	Игра престолов	Друзья	Доктор Кто	Как я встретил вашу маму
Любимый завтрак			манная каша	суп	бутерброды с колбасой
Хобби	гитара		кроссворды	йога	компьютерные игры

Далее учтем утверждения

- ✓ Сосед того, кто ест мюсли на завтрак, любит играть на гитаре.
- ✓ Сосед того, кто ест яичницу на завтрак, любит читать.
- ✓ Тирион ест на завтрак яичницу.

Получаем:

- ✓ Мюсли едят во второй палатке.
- ✓ В первой палатке живет Тирион, как любитель яичницы. Значит, Джон во второй палатке.
- ✓ Джон любит читать.

Итак, таблица заполнена.

Номер	1	2	3	4	5
Имя	Тирион	Джон	Серсея	Петрович	Дейнерис

Цвет палатки	белый	зеленый	красный	синий	желтый
Любимый сериал	Очень странные дела	Игра престолов	Друзья	Доктор Кто	Как я встретил вашу маму
Любимый завтрак	яичница	мюсли	манная каша	суп	бутерброды с колбасой
Хобби	гитара	чтение	кроссворды	йога	компьютерные игры

Еще раз проверим, что все утверждения верны.

- ✓ Известно, что Серсея живет в красной палатке.
- ✓ Дейенерис смотрит сериал «Как я встретил вашу маму».
- ✓ Палатка Дейенерис стоит сразу справа от палатки Петровича.
- ✓ Предпочитающий есть на завтрак манную кашу любит отгадывать кроссворды.
- ✓ Белая палатка имеет номер 1.
- ✓ Сосед того, кто ест мюсли на завтрак, любит играть на гитаре.
- ✓ Сосед того, кто ест яичницу на завтрак, любит читать.
- ✓ Тот, кто любит на завтрак суп, смотрит сериал «Доктор Кто».
- ✓ Зеленая палатка стоит рядом с той, в которой смотрят сериал «Очень странные дела».
- ✓ Джон терпеть не может суп на завтрак.
- ✓ Серсея живет между зеленой и синей палатками.
- ✓ На гитаре играют по соседству с зеленой палаткой.
- ✓ Житель синей палатки занимается йогой.
- ✓ Тот, кто любит компьютерные игры, живет по соседству с увлекающимся йогой.
- ✓ В центральной палатке (под номером 3) смотрят сериал «Друзья».
- ✓ Тирион ест на завтрак яичницу.
- ✓ Обитатель палатки зеленого цвета смотрит сериал «Игра престолов».
- ✓ Живущий в желтой палатке ест на завтрак бутерброды с колбасой.

Итак, Тирион живет в первой слева палатке, белого цвета, смотрит сериал «Очень странные дела», ест на завтрак яичницу и любит играть на гитаре.

Задача является вариацией знаменитой «Задачи Эйнштейна», хотя нет доказательств, что она действительно придумана великим ученым. Прочитать о ней можно, например, в Википедии, или здесь <https://newgoal.ru/zadacha-ejnshtejna-pro-5-domov/>.

Возможно, вам удалось найти другое решение, не противоречащее условиям.

В остальных вариантах приведены задачи такого же типа.

Ответ. Тирион живет в первой слева палатке, белого цвета, смотрит сериал «Очень странные дела», ест на завтрак яичницу и любит играть на гитаре.

4. Обработка больших массивов данных

Задача 4. Во время проведения отчета по работе скважины произошел сбой при передаче данных по ежедневному объему добычи газа. Отчет состоит из собственно данных и контрольного числа SUM . Это есть максимальное число, которое удовлетворяет следующим условиям:

- 1) $SUM = A + B$, где A и B — объем добычи в различные два дня, при этом объем добычи в первый из этих дней больше, чем объем добычи во второй из этих дней;
- 2) SUM делится на 10;
- 3) произведение A и B не кратно шести.

Если контрольного значения, удовлетворяющего условиям, нет, то значения не корректны и отчет не сформирован.

Напишите программу, проверяющую корректность контрольного значения. Обратите внимание, что количество дней может быть очень велико, а под переменные выделяется не более 1 Кб памяти.

Кратко опишите используемый алгоритм решения.

Входные данные

На вход программе в первой строке подается количество дней N ($1 < N < 5000$), далее в N строках подаются объемы добычи газа (натуральные числа, не превышающие 500). В последней строке записано контрольное значение.

Выходные данные

Программа должна напечатать отчёт по следующей форме:

Контрольное значение:

Отчет сформирован. (или — Отчет не сформирован)

Пример.

Входные данные

10
300
405
412
200
135
105
35
408
110
260
540

Выходные данные

Контрольное значение: 540

Отчет сформирован.

Примечание. Здесь $A=405$, $B=135$.

Решение задачи 4.

Обратите внимание – если в массив сохранить все данные, то при заданных ограничениях по числу дней (не более 5000) занимаемая память превысит заданный объем в 1 Кб. Такие решения оценивались максимум на 7 баллов, тем не менее, они возможны. Можно перебрать все пары элементов массива, соответствующие заданным условиям, и выбрать пару с максимальной суммой.

Первая идея решения. Рассматриваем последовательность целых чисел. Требуется найти максимальную сумму $SUM=A+B$, то есть, она должна получаться из максимально возможных слагаемых A и B , соответствующих условиям. Договоримся, что слагаемое A появляется раньше B . Тогда условия принимают вид:

- 1) $A > B$,
- 2) $AB \bmod 6 > 0$,
- 3) $(A + B) \bmod 10 = 0$.

Пусть пришло очередное число B . Чтобы выполнялось условие 2 требуется $B \bmod 6 > 0$. Если это не так, B не может участвовать в формировании контрольной суммы. Пусть $B \bmod 6 > 0$, тогда для соблюдения условия 2 требуется следующее:

- Если $B \bmod 2 = 0$, то $A \bmod 3 > 0$;
- Если $B \bmod 3 = 0$, то $A \bmod 2 > 0$;
- Если два условия выше не выполнены, то A может быть любым, но не кратным 6.

Для соблюдения условия 3 слагаемое A должно удовлетворять ограничению

$$((A \bmod 10) + (B \bmod 10)) \bmod 10 = 0.$$

Кроме того, A должно быть максимальным. Итак, имеет смысл завести два массива (списка) размерности 10, $A2[]$ и $A3[]$, и хранить в них максимальные числа, **не кратные** соответственно 3 и 2, с соответствующими остатками от деления на 10. Если число кратно 6, оно не попадет ни в один массив. Когда приходит очередное число B , подбираем ему соответствующую пару из массивов:

- Вычисляем остаток $j = B \bmod 10$ и дополнительный остаток $p = (10 - j) \bmod 10$, число A должно иметь остаток от деления на 10, равный p ;
- Если B не кратно 2, в пару с ним можно взять число $A2[p]$, при этом B может быть как кратно, так и не кратно 3;
- Если B не кратно 3, в пару с ним можно взять число $A3[p]$, при этом B может быть как кратно, так и не кратно 2.

Проверяем для подобранных пар условие 1. Если оно верно, обновляем максимальную сумму. Затем обновляем массивы $A2[]$ и $A3[]$, используя B .

Ниже приведена программа на языке Pascal.

```
Program Problem_4;
```

```
var B,i,j,p,n,sum,maxsum:integer;
```

```

    A2,A3:array[0..9] of integer;
begin
    sum:=-1;
    maxsum:=-1;
    readln(n);
    for i:=0 to 9 do begin
        A2[i]:=0;
        A3[i]:=0;
    end;
    for i:=1 to n do begin
        readln(B);
        j:=B mod 10;
        p:=(10-j)mod 10;
        if B mod 2 > 0 then
            if (A2[p]>B) and (A2[p]+B> sum) then
                sum:=A2[p]+B;
        if B mod 3 > 0 then
            if (A3[p]>B) and (A3[p]+B> sum) then
                sum:=A3[p]+B;
        if (sum>maxsum) then maxsum:=sum;
        if (B mod 2>0) and (B>A2[j]) then A2[j]:=B;
        if (B mod 3>0) and (B>A3[j]) then A3[j]:=B;
    end;
    writeln ('Контрольное значение: ', maxsum);
    readln(B);
    if ((B=maxsum) and (maxsum>-1)) then
        writeln ('Отчет сформирован')
    else
        writeln ('Отчет не сформирован');
end.

```

Ниже приведена программа на языке C++.

```

#include <iostream>
using namespace std;
int main()
{

```

```

int B,i,j,p,n,sum,maxsum,A2[10],A3[10];
sum=maxsum=-1;
cin >> n;
for(i=0;i<10;i++)
    A2[i]=A3[i]=0;
for(i=1;i<=n;i++)
{
    cin>>B;
    j=B%10;
    p=(10-j)%10;
    if(B%2>0)
        if((A2[p]>B)&&(A2[p]+B>sum))
            sum=A2[p]+B;
    if(B%3>0)
        if((A3[p]>B)&&(A3[p]+B>sum))
            sum=A3[p]+B;
    if(sum>maxsum) maxsum=sum;
    if((B%2>0)&&(B>A2[j])) A2[j]=B;
    if((B%3>0)&&(B>A3[j])) A3[j]=B;
}
cout << "Контрольное значение: " << maxsum << endl;
cin >> B;
if ((B==maxsum)&&(maxsum>-1))
    cout<< "Отчет сформирован"<<endl;
else
    cout<< "Отчет не сформирован"<<endl;
return 1;
}

```

Ниже представлена программа на языке Python.

```

A2=[0]*10
A3=[0]*10
sum=-1
maxsum=-1
n=int(input())

```

```

for i in range(n):
    B=int(input())
    j=B%10
    p=(10-j)%10
    if B%2>0:
        if A2[p]>B and A2[p]+B>sum:
            sum=A2[p]+B
    if B%3>0:
        if A3[p]>B and A3[p]+B>sum:
            sum=A3[p]+B
    if sum>maxsum:
        maxsum=sum
    if B%2>0 and B>A2[j]:
        A2[j]=B
    if B%3>0 and B>A3[j]:
        A3[j]=B
print("Контрольное значение: ", maxsum)
B=int(input())
if B==maxsum and maxsum>-1:
    print("Отчет сформирован")
else:
    print("Отчет не сформирован")

```

Вторая идея решения. Заметим, что числа находятся в диапазоне от 1 до 500. Поэтому для информации о числах, поступивших ранее, достаточно иметь массив из 500 переменных, принимающих два значения: 0 (если число на вход не поступало) и 1 (если поступало), для хранения такой переменной достаточно одного байта, то есть, например, подойдет логический тип данных, символьный или короткое целое. Тогда, получив на вход очередное число B , мы проверим числа, большие B и поступившие до него, на соответствие условиям, и, при необходимости, обновим максимальную сумму. Этот процесс можно организовать по-разному, от полного перебора всех чисел до эффективного перебора. Эффективнее всего будет найти максимальное число, удовлетворяющее условиям, то есть просматривать числа от большего к меньшему, и выбирать только те числа, которые в сумме с B делятся на 10 без остатка, то есть перебирать числа с шагом 10. Тогда останется проверить только условие на произведение чисел.

Ниже приведена программа на языке Pascal.

```
Program Problem_4;
var B,i,j,k,maxsum,n:integer;
A:array[1..500] of byte;
begin
maxsum:=-1;
for i:=1 to 500 do A[i]:=0;
readln(n);
for i:= 1 to n do begin
  readln(B);
  if(B mod 6 = 0) then continue;
  j:=B mod 10;
  k:=500-j;
  while (k>B) do begin
    if (A[k]=1) and (k*B mod 6 > 0) then begin
      if (k+B>maxsum) then maxsum:=k+B;
      break;
    end;
    k:=k-10;
  end;
  A[B]:=1;
end;
writeln ('Контрольное значение: ', maxsum);
readln(B);
if ((B=maxsum) and (maxsum>-1)) then
  writeln ('Отчет сформирован')
else
  writeln ('Отчет не сформирован');
end.
```

Ниже приведена программа на языке C++.

```
#include <iostream>
using namespace std;
int main()
{
```

```

int B,i,j,p,n,maxsum;
bool A[501];
maxsum=-1;
cin >> n;
for(i=0;i<501;i++)
    A[i]=0;
for(i=1;i<=n;i++)
{
    cin>>B;
    if(B%6==0) continue;
    j=B%10;
    p=500-j;
    while(p>B)
    {
        if((A[p])&&(p*B%6>0))
        {
            if (p+B>maxsum) maxsum=p+B;
            break;
        }
        p-=10;
    }
    A[B]=1;
}
cout << "Контрольное значение: " << maxsum << endl;
cin >> B;
if ((B==maxsum)&&(maxsum>-1))
    cout<< "Отчет сформирован"<<endl;
else
    cout<< "Отчет не сформирован"<<endl;
return 1;
}

```

5. Графы

Задача 5. Для подготовки к возведению буровой установки могут проводиться следующие работы:

Обозначение работы	Наименование работы	Профессии исполнителей
Подготовительные работы к возведению буровой установки		
а.	Определение на местности точки заложения скважины	Топографы
б.	Выкорчевка леса	Бульдозеристы
в.	Расчистка площадей для строительства подъездного пути	Бульдозеристы
г.	Строительство подъездного пути	Дорожники
д.	Подвод электроэнергии	Электрики
е.	Подвоз монтажной бригады	Водители
ж.	Сооружение временного лагеря для монтажной бригады	Монтажники
з.	Организация снабжения водой бытовых и производственных объектов	Сварщики
и.	Наладка связи	Электрики
к.	Подвоз материалов для строительства жилого поселка	Водители
л.	Монтаж конструкций жилого поселка	Монтажники
м.	Расчистка и планировка площадки для строительства БУ	Бульдозеристы
н.	Организация снабжения электроэнергией бытовых и производственных объектов	Электрики

Каждая работа требует определенного количества человеко-дней. Также для каждой работы имеется список предшествующих работ (который может быть пустым). Работа

начинается на следующий день после того, как были закончены все предшествующие работы. Информация по работам на месторождении Дальнем сведена в таблицу.

Обозначение работы	Наименование работы	Обозначение предшествующих работ	Объем работы, чел.дн
1	2	3	4
г.	Строительство подъездного пути		182
д.	Подвод электроэнергии		40
ж.	Сооружение временного лагеря для проживания монтажной бригады	г	156
з.	Организация снабжения водой бытовых и производственных объектов	л	104
л.	Монтаж конструкций жилого поселка	д, ж	260
н.	Организация снабжения электроэнергией бытовых и производственных объектов	д	65

Численность имеющегося персонала приведена в таблице.

Профессия исполнителей	Наличная численность персонала, чел.	Профессия исполнителей	Наличная численность персонала, чел.
Топографы	3	Водители	29
Бульдозеристы	10	Сварщики	8
Дорожники	12	Подготовительная бригада	6
Монтажники	66	Приемочная комиссия	12
Электрики	12		

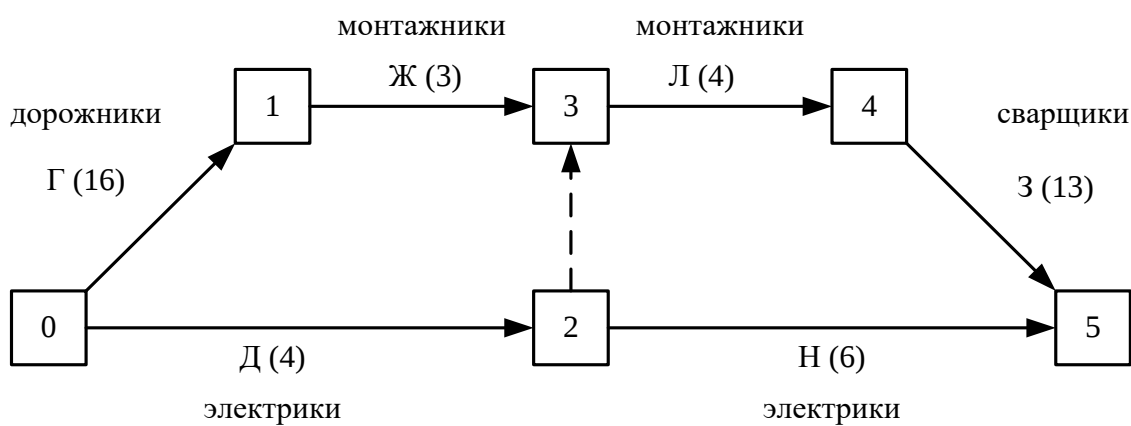
Определите минимальный срок, за который можно закончить подготовительные работы на месторождении Дальнем. Сколько сотрудников каждой профессии при этом понадобится?

Решение задачи 5. Рассчитаем сначала, какое минимальное количество дней тратится на каждую работу. Для этого поделим объем работы на количество исполнителей. Учтем, что работы начинаются на следующий день после завершения всех предшествующих работ. Это означает, что количество дней надо округлить сверху до ближайшего целого числа. Результаты сведем в таблицу.

Обозначение работы	Наименование работы	Профессии исполнителей	Объем работы, чел./дн	Кол-во исполнителей	Кол-во дней
1	2	3	4	5	6
г.	Строительство подъездного пути	Дорожники	182	12	16
д.	Подвод электроэнергии	Электрики	40	12	4
ж.	Сооружение временного лагеря для проживания монтажной бригады	Монтажники	156	66	3
з.	Организация снабжения водой бытовых и производственных объектов	Сварщики	104	8	13
л.	Монтаж конструкций жилого поселка	Монтажники	260	66	4
н.	Организация снабжения электроэнергией бытовых и производственных объектов	Электрики	65	12	6

Связь между работами покажем на сетевом графике. Здесь вершины графа соответствуют этапам начала и окончания работ, ребра соответствуют работам, возле ребер указаны исполнители, обозначения работ, в скобках – минимальное число дней, которое тратится на работу.

Здесь вершина 0 – начало всех работ, из нее выходят ребра Г и Д, так как эти работы не требуют предшествующих работ. Вершина 1 – окончание работы Г, вершина 2 – окончание работы Д. Так как работе Ж предшествует работа Г, соответствующее ребро выходит из вершины 1, вершина 3 – окончание работы Ж. Далее рассмотрим работы Л и Н. Работе Н предшествует работа Д, т.е., началом соответствующего ребра будет вершина 2. Работа Н не предшествует ни одной работе, так что ее окончание совпадет с окончанием всех работ, это вершина 5. Работе Л предшествуют работы Ж и Д, так что можно начать соответствующее ребро в вершине 3, а также провести ребро (2,3) (пунктирная линия). Оно не соответствует никакой работе, и нужно лишь, чтобы показать порядок предшествования. Работа Л предшествует работе 3, а работа 3 не предшествует никакой работе, т.е., ее окончание совпадает с окончанием всех работ.



Проанализируем граф. Нужно рассмотреть пути от вершины 0 до вершины 5 и найти самый длинный (по времени) путь. Здесь это 01345, т.е., надо постараться минимизировать именно его. Заметим, что специалисты одной профессии никогда не работают параллельно на двух работах, монтажники сначала завершают работу Ж, потом переходят к работе Л, т.е., не требуется делить персонал между работами. Постараемся выполнить все работы на пути в кратчайшие сроки, указанные в скобках. Найдем необходимое для этого количество исполнителей: поделим число человеко-дней на число дней и округлим сверху до целого числа. Результаты сведем в таблицу. В последней колонке указано количество исполнителей. Видно, что не имеет смысла, например, задействовать 66 монтажников для работы Ж, так как 52 монтажника выполнят эту работу за те же три дня. Всего работы займут $16+3+4+13=36$ дней.

Обозн. работы	Наименование работы	Профессии исполнителей	Объем работы, чел./дн	Кол-во дней	Кол-во исполнителей
1	2	3	4	5	6
Г.	Строительство подъездного	Дорожники	182	16	12

Обозн. работы	Наименование работы	Профессии исполнителей	Объем работы, чел./дн	Кол-во дней	Кол-во исполнителей
1	2	3	4	5	6
	пути				
ж.	Сооружение временного лагеря для проживания монтажной бригады	Монтажники	156	3	52
з.	Организация снабжения водой бытовых и производственных объектов	Сварщики	104	13	8
л.	Монтаж конструкций жилого поселка	Монтажники	260	4	65

Рассмотрим теперь путь 025. Данные работы выполняются по очереди электриками. Работа Д должна быть завершена к окончанию работы Ж, чтобы можно было начать работу Л, т.е., за $16+3=19$ дней. Затем работу Н нужно завершить к окончанию работы З, т.е., за $4+13=17$ дней. Рассчитаем необходимое число исполнителей.

Обозн. работы	Наименование работы	Профессии исполнителей	Объем работы, чел./дн	Кол-во дней	Кол-во исполнителей
1	2	3	4	5	6
д.	Подвод электроэнергии	Электрики	40	19	3
н.	Организация снабжения электроэнергией бытовых и производственных объектов	Электрики	65	17	4

Таким образом, работы займут 36 дней, численность исполнителей на каждом этапе указана в таблице.

Обозн. работы	Наименование работы	Профессии исполнителей	Кол-во исполнителей
1	2	3	6
г.	Строительство подъездного пути	Дорожники	12
ж.	Сооружение временного лагеря для проживания монтажной бригады	Монтажники	52
з.	Организация снабжения водой бытовых и производственных объектов	Сварщики	8
л.	Монтаж конструкций жилого поселка	Монтажники	65
д.	Подвод электроэнергии	Электрики	3
н.	Организация снабжения электроэнергией бытовых и производственных объектов	Электрики	4

Более подробно о сетевых графиках рассказано в работе Заведеева Е.В., отсюда же взяты и данные к заданиям.

Заведеев, Е.В. Применение сетевого планирования и управления на предприятиях нефтяной и газовой промышленности: учеб.-метод. пособие / Е.В. Заведеев. – Сургут: ИЦ СурГУ, 2009. – 64 с.

Это лишь один из способов решения данной задачи. Возможно, вы нашли свой вариант.

Задачи в остальных вариантах аналогичны.

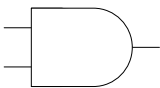
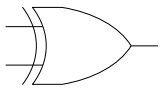
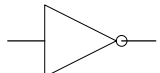
6. Синтез дискретных устройств

Задача 6. На месторождении Дальнем произошел отказ оборудования. Выяснилось, что сгорело небольшое, но важное устройство. Необходимо срочно его восстановить. У геологов Иннокентия и Серсеи есть набор базовых элементов, а также таблица истинности, описывающая данное устройство (a, b, c – входы, y – выход). Что, если это не последняя поломка? Помогите геологам собрать две схемы, реализующие данное устройство, либо подскажите, какие элементы им еще понадобятся.

Таблица истинности:

a	b	c	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Базовые элементы.

Название	Изображение	Количество
И (с двумя входами)		8
Исключающее ИЛИ (с двумя входами)		3
Инвертор		5

Решение задачи 1. Сначала следует получить формулу для функции $y(a, b, c)$, которая содержит только инверсии, конъюнкции и сумму по модулю два. Замечаем, что при $a = 0$ (верхняя половина таблицы) столбец значений совпадает со столбцом значений суммы по

модулю два $b \oplus c$, а при $a = 1$ (нижняя половина таблицы) единицы соответствуют строкам, где $c = 1$. Отсюда следует формула

$$y = (\bar{a} \wedge (b \oplus c)) \vee (a \wedge c).$$

Убедиться, что она верна, можно с помощью таблицы истинности. По условию задачи, не имеется элемента ИЛИ, и формула содержит дизъюнкцию.

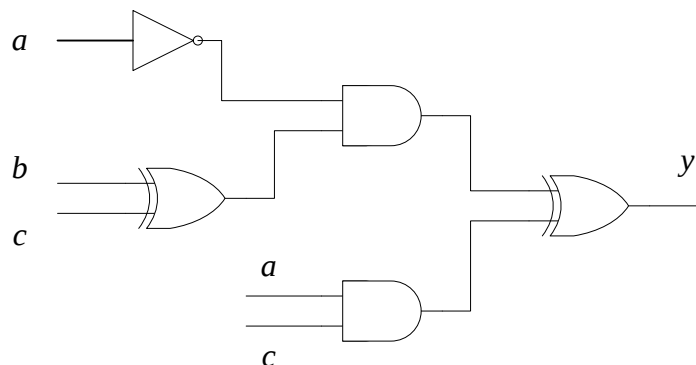
Заметим, что слагаемые в дизъюнкции, а именно, $(\bar{a} \wedge (b \oplus c))$ и $(a \wedge c)$, не равны единице одновременно. В этом случае функция «или» равносильна функции «исключающее или», и формулу можно переписать

$$y = (\bar{a} \wedge (b \oplus c)) \oplus (a \wedge c).$$

Построим таблицу истинности для этой формулы.

a	b	c	$\bar{a}$	$b \oplus c$	$\bar{a} \wedge (b \oplus c)$	$(a \wedge c)$	$(\bar{a} \wedge (b \oplus c)) \oplus (a \wedge c)$
0	0	0	1	0	0	0	0
0	0	1	1	1	1	0	1
0	1	0	1	1	1	0	1
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	0	1	0	1	1
1	1	0	0	1	0	0	0
1	1	1	0	0	0	1	1

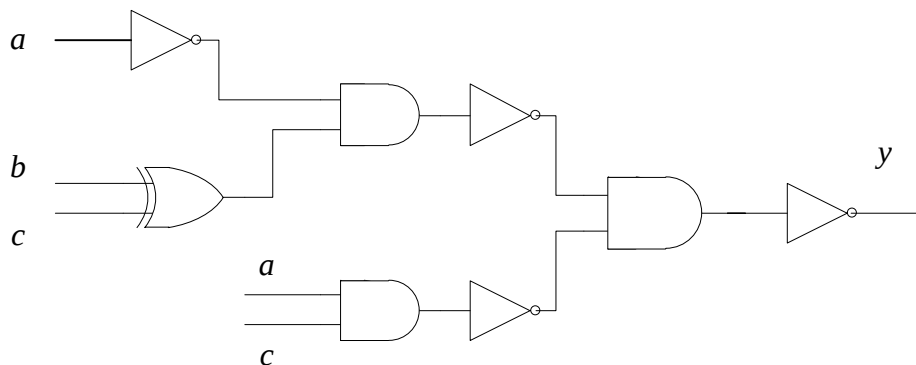
Для реализации этой формулы необходим один инвертор, два элемента «исключающее ИЛИ», два элемента «И», т.е., достаточно имеющихся элементов. Схема имеет вид:



После сборки данной схемы осталось четыре инвертора, один элемент «исключающее или», шесть конъюнкторов, то есть, вторую схему нужно строить по другой формуле. Можно преобразовать исходную формулу, применив закон де Моргана $a \vee b = \overline{\bar{a} \wedge \bar{b}}$:

$$y = (\bar{a} \wedge (b \oplus c)) \vee (a \wedge c) = \overline{(\bar{a} \wedge (b \oplus c))} \wedge \overline{(a \wedge c)}$$

Для построения этой схемы нужно четыре инвертора, три конъюнктора и один элемент «исключающее или», то есть, схему построить можно.



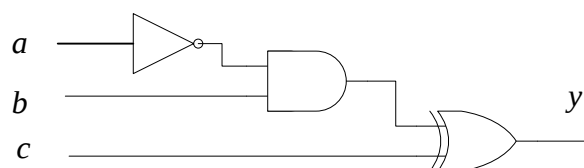
Можно найти совершенно другую формулу, если заметить, что $y = c$ везде, кроме случая, когда $a = 0, b = 1$, в этом случае $y = \bar{c}$. Учтем, что $x \oplus 0 = x, x \oplus 1 = \bar{x}$. Это приводит к следующей формуле:

$$y = (\bar{a} \wedge b) \oplus c.$$

Убедимся, что она верна, построив таблицу истинности.

a	b	c	$\bar{a}$	$(\bar{a} \wedge b)$	$(\bar{a} \wedge b) \oplus c$
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	0	0
1	1	1	0	0	1

Схема имеет вид (имеющихся элементов достаточно).



В остальных вариантах задачи аналогичны, приводится решение только для одного варианта.

7. Комбинаторика

Задача 7. На случай отсутствия связи у геологоразведочных групп имеются сигнальные ракеты. Сигналом являются четыре, либо три выстрела подряд. У группы имеются четыре ракеты с зарядами разного цвета: два белых, зеленый и красный. Каждая ракета используется не более одного раза. С помощью последовательности выстрелов разных цветов могут быть переданы различные сообщения. Какое максимальное число сообщений можно закодировать, соблюдая следующие три правила одновременно:

- 1) В последовательности из трех сигналов все сигналы имеют разные цвета;
- 2) Последовательность из трех сигналов не начинается с зеленого сигнала;
- 3) Никакая последовательность сигналов не является началом другой последовательности.

Решение задачи 7. Поскольку последовательностей длины 4 существует больше, чем длины 3, рассмотрим сначала все возможные последовательности длины 4, удовлетворяющие условиям. Последовательность не начинается с сигнала зеленого цвета, поэтому первый сигнал можно выбрать двумя способами: красный и белый. Второй сигнал будет зависеть от первого: так как красный сигнал один, его нельзя повторить. После красного сигнала идут два белых и зеленый, всего будет три варианта (по числу позиций, на которые можно поставить зеленый сигнал):

КББЗ

КБЗБ

КЗББ

Если первый сигнал белый, то после него идут белый, зеленый и красный, всего имеется шесть перестановок этих сигналов (3!).

ББЗК

ББКЗ

БКЗБ

БКБЗ

БЗКБ

БЗБК

Рассмотрим теперь последовательности длины 3. Они не должны совпадать с началом последовательности длины 4, поэтому можно начать с зеленого сигнала. После него могут идти белый и красный в любом порядке, всего два варианта.

ЗБК

ЗКБ

Ответ. 11 сообщений.

8. Кодирование и декодирование

Задача 8. Бриенна устраивается на работу в отдел защиты информации компании Восточный Газ. На собеседовании ей предложили разгадать небольшую криптограмму. Дана таблица и последовательность цифр, необходимо соотнести буквы и цифры.

К	*	ИЯ	=	ЗК
+		*		+
ИА	*	ИА	=	ККА
=		=		=
ИЦ	+	КГФ	=	КАЦ

0	1	2	3	4	5	6	7

45310125716 – закодированное слово.

Необходимо сопоставить цифры с буквами и составить закодированное слово.

Решение задачи 8. Решение нужно начать с анализа имеющийся таблицы и попробовать найти самые простые связи в строке или столбце. Во-первых, у нас в таблице не будет 8 и 9. В третьей строке Ц + Ф дают ответ Ц. Отсюда вывод, что Ф – это 0.

Следующей строкой является ИА * ИА = ККА. Т.е. тут квадрат числа дает такую же цифру на конце и одинаковые сотню и десяток. Таких числа есть всего два: 225 и 441.

Проверим вариант с 225, если дальше будут противоречия – возьмем 441. Значит И это 1, А это 5, а К это 2.

2	*	1Я	=	(3)2
+		*		+
15	*	15	=	225

=		=		=
1Ц	+	2Г0	=	25Ц

Частично заполняем таблицу. В первом столбце $2 + 15 = 17$, следовательно Ц – это 7. Во втором столбце $17 + 2Г0 = 257$, отсюда Г=4.

Далее получаем что Я это 6, а 3 это 3.

2	*	16	=	32
+		*		+
15	*	15	=	225
=		=		=
17	+	240	=	257

Далее вносим все буквы в таблицу

Ф	И	К	З	Г	А	Я	Ц
0	1	2	3	4	5	6	7

Все что осталось сделать это записать буквы в том порядке что и в коде 45310125716 = ГАЗИФИКАЦИЯ

Ответ: ГАЗИФИКАЦИЯ

9. Логика

Задача 9. Тирион приехал на смену в Норильск, но от холода забыл большую часть информации, связанную с этой сменой. Смена может начаться в понедельник или вторник, днем или ночью, с Ивановым или с Петровым, в офисе пункта приема металла или в офисе напротив. Три мужика, которых он повстречал на вахте, решили над ним подшутить. Каждый высказал свою версию смены Тириона:

Слова первого: “Его смена начинается в понедельник, на работу должен выйти днём”.

Слова второго: “Работает с Ивановым, рабочее место находится в офисе напротив пункта приема металла”.

Слова третьего: “Его смена начинается в понедельник, рабочее место - в офисе пункта приема металла”

Управляющий, который никогда не врет, поддержал шутку, но дал подсказку: один из мужиков не соврал ни разу, второй обманул оба раза, третий сказал, как правду, так и ложь. Определите: в какой день недели, в какое время суток, с кем и где будет работать сотрудник.

Примечание: у Петрова в понедельник выходной.

Решение задачи 9.

Для решения задачи необходимо рассмотреть различные варианты и отсеять неподходящие.

1. Первый – не соврал, второй – соврал один раз, третий – соврал. Но посмотрев на высказывания, можно заметить, что данный вариант не подойдет, потому что первый и третий говорят о дне работы, значит будет противоречие.

2. Первый – не соврал, второй – соврал, третий – соврал один раз. Значит Тирион будет работать в понедельник днем в офисе напротив пункта приема металла с Петровым. Но будет противоречие в рабочем месте, так как по словам второго мужика рабочее место будет в пункте приема металла.

3. Первый – соврал один раз, второй – не соврал, третий – соврал. Тогда Тирион будет работать с Ивановым в офисе напротив пункта приема металла во вторник (тогда у первого ложь будет насчет дня недели) днем – этот вариант подходит.

4. Первый – соврал один раз, второй – соврал, третий – не соврал. Тогда получается, что Петров должен работать в понедельник, что противоречит примечанию.

5. Первый – соврал, второй – соврал один раз, третий – не соврал. Аналогично первому варианту, посмотрев на высказывания, можно заметить, что данный вариант не подойдет, потому что первый и третий говорят о дне работы, значит будет противоречие.

6. Первый – соврал, второй – не соврал, третий – соврал один раз. Если предположить, что третий соврал насчет дня недели, тогда будут противоречия в рабочем месте. Если же наоборот, то противоречие возникнет в дне недели.

Ответ: Тирион будет работать с Ивановым в офисе напротив пункта приема металла во вторник днем.

10. Кодирование и декодирование

Задача 10. У нас сломался аппарат дешифровки сообщений, но нам срочно нужно узнать, что в сообщении, которое мы только что получили. Всё чем мы можем тебе помочь, это дать тестовый набор данных. Если передать слово СВОБОДА, то на другом конце получат сообщение ВРРУЖГД.

Тестовый набор:

СВОБОДА

ВБДСООА

ДГЖУРРВ

ВРРУЖГД

зашифрованный текст:

ХПБФНЦФБЗЗВРПИРРМ ЛЮЕПТФТ

Примечание. Буквы Ё и Й в алфавите не используются, а пробел стоит перед буквой А в алфавите. Регистр не имеет значения.

Решение задачи 10.

Для начала нужно разгадать алгоритм шифрования. Посмотрим на представленный пример:

СВОБОДА – первоначальное слово.

ВБДСООА – далее сначала ставятся буквы на четных местах, затем на нечетных.

ДГЖУРРВ – сдвиг по алфавиту вправо на 2 символа.

ВРРУЖГД – слово переворачивается.

Для расшифровки нужно выполнить все в обратном порядке:

ХПБФНЦФБЗЗВРПИРРМ ЛЮЕПТФТ

Перевернем:

ТФТПЕЮЛ МРРИПРВЗЗБФЦНФБПХ

Сдвигаем на 2 символа влево:

РТРНГЬИЮКООЖНОАЕЕ ТЧЛТ НУ

Посчитаем количество букв и разделим слово по середине, и теперь буквы сначала ставим на четные позиции:

РТРНГЬИЮКООЖ|НОАЕЕ ТЧЛТ НУ

Р Т Р Н Г Ъ И Ю К О О Ж

Теперь вторую половину слов ставим на нечетные места:

НРОТАРЕНЕГ ЪТИЧЮЛКТО ОНЖУ

Ответ в таком виде считается правильным. При формировании билетов была допущена ошибка и изначальная строка была перевернута лишней раз, поэтому связанной фразы не получится. Чтобы текст стал осмысленным конечный ответ нужно перевернуть, перед этим поставив первую букву в конец: НУЖНО ОТКЛЮЧИТЬ ГЕНЕРАТОР

Ответ: Нужно отключить генератор