

**Методические указания по Отраслевой олимпиаде школьников
«Газпром», профиль «информатика»**

Учебное пособие для подготовки к олимпиаде

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

Тематика заданий

1. Синтез дискретных устройств
2. Жадные алгоритмы
3. Логика
4. Обработка большого массива данных
5. Графы
6. Шифрование
7. Комбинаторика
8. Кодирование и декодирование
9. Передача информации
10. Системы счисления

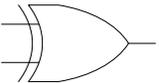
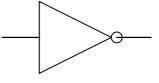
1. Синтез дискретных устройств

Задача 1. 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 разработать наиболее простую схему устройства, содержащую как можно меньше элементов, либо подскажите, какой набор элементов им нужен.

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.

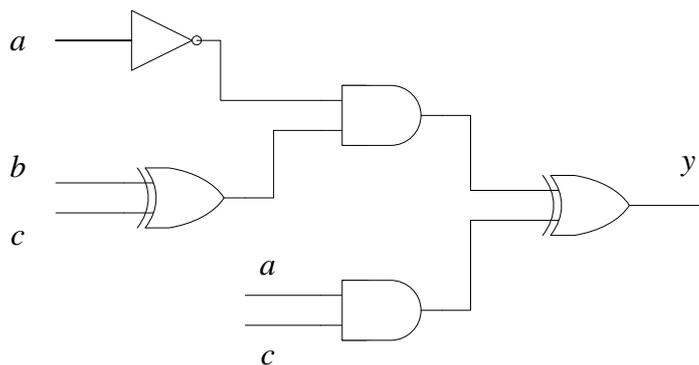
Заметим, что слагаемые в дизъюнкции, а именно, $(\bar{a} \wedge (b \oplus c))$ и $(a \wedge c)$, не равны единице одновременно. В этом случае функция «или» равносильна функции «исключающее или», и формулу можно переписать

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

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:



Двоим участникам удалось найти более простые решения:

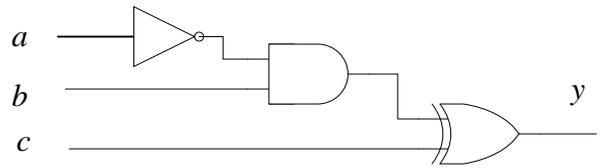
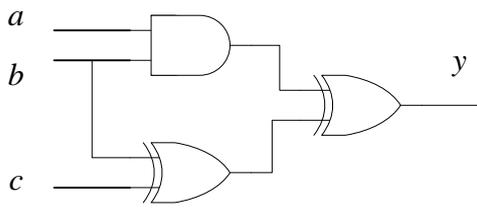
$$y = (a \wedge b) \oplus (b \oplus c),$$

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

Убедимся, что они верны, построив таблицу истинности.

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

Схемы имеют вид



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

2. Жадные алгоритмы

Задача 2.1. Имеется список k скважин, принадлежащих компании Дотракия, на которых требуется провести геолого-технические мероприятия. Известно количество дней, которые требуются для проведения мероприятия на каждой скважине. Все работы должны быть завершены за N дней. Предложите алгоритм расчета, какое максимальное количество скважин можно обслужить за выделенное время. Напишите программу, реализующую данный алгоритм.

Входные данные

В первой строке вводятся число N (натуральное, не превышает 100), и число k (натуральное, не превышает 50). Затем идет k чисел – количество дней, которые требуются, чтобы провести мероприятия на скважинах (времена – натуральные числа, не превосходят 100).

Выходные данные

Выведите единственное число – максимальное количество скважин, на которых можно провести мероприятия за заданное время.

Пример.

Входные данные

17 3

12 4 10

Выходные данные

2

Примечание. Можно провести мероприятия на первой и второй, либо на второй и третьей скважинах.

Решение задачи 2.1. Данную задачу можно решить жадным алгоритмом.

Жадные алгоритмы не смотрят в будущее, чтобы выбрать глобальное оптимальное решение. Их интересует только лучшее решение в данный момент. В общем случае оптимальное решение может отличаться от решения, которое выбирает алгоритм на каждом шаге своей работы, поэтому жадным алгоритмом можно найти точное решение только очень ограниченного класса задач. Однако, обычно они являются вычислительно простыми, в связи с чем используются для поиска приближенных решений сложных задач.

Чтобы более подробно изучить жадные алгоритмы, можно, например, посмотреть следующие ролики:

<https://www.youtube.com/watch?v=ccb9NCGTDk>

https://youtu.be/yXYR_JuojdY

Также можно рекомендовать презентации по жадным алгоритмам:

<http://cs.mipt.ru/wp/wp-content/uploads/2018/09/greedyalgs.pdf>

<https://studfile.net/preview/3276187/>

Поскольку жадные алгоритмы широко используются для задач оптимизации, по этой теме имеется много литературы, которую можно найти самостоятельно.

Вернемся к нашей задаче. Чтобы провести мероприятия на как можно большем скважин, нужно выбирать скважины, на которые тратится меньше всего времени. Тогда останется больше времени на работу на остальных скважинах. Другими словами, если в плане мероприятий заменить скважину с продолжительностью мероприятий X дней, на скважину с продолжительностью мероприятий Y дней, при $X > Y$, то мы выиграем $X - Y > 0$ дней, и количество скважин, на которых мы можем провести мероприятия, может только увеличиться. Таким образом, алгоритм может быть таким:

1. Записываем количество дней, выделенных на все мероприятия, в переменную n .
2. Сохраняем данные по времени обслуживания в массив A .
3. Обнуляем счетчик $count$
4. Находим минимальный элемент в массиве A , меньший n , и запоминаем его номер j . Если такого элемента нет, выводим значение $count$ и заканчиваем.
5. Если такой элемент нашелся, увеличиваем $count$ на единицу, уменьшаем n на величину $A[j]$ и присваиваем $A[j]$ значение, заведомо большее n (чтобы не выбрать тот же элемент в качестве минимума в следующий раз).
6. Идем на шаг 4.

Ниже приведена программа на языке Pascal.

```
Program Problem_2_1;
var n, k, count, min, i, j: integer;
    A: array[1..50] of integer;
begin
    readln(n, k);
    for i:= 1 to k do
        read(A[i]);
    min:=0;
    count:=0;
    while (min<1000) do
        begin
            min:=1000;
            for i:=1 to k do
                if ((A[i]<=n) and (A[i]<min)) then
                    begin
                        j:=i;
                        min:=A[i];
                    end
                end
        end
```

```

        end;
        if (min<=n) then
        begin
            n:=n-min;
            count:=count+1;
            A[j]:=1000;
        end;
    end;
    writeln(count);
end.

```

Ниже приведена программа на языке C++.

```

#include<iostream>
using namespace std;

int main()
{
    int n,k,count,min,i,j,A[50];
    cin>> n >> k;
    for(i=0;i<k;i++)
    cin>> A[i];
    min=0;
    count=0;
    while(min<1000)
    {
        min=1000;
        for(i=0;i<k;i++)
            if((A[i]<=n) && (A[i]<min))
            {
                j=i;
                min=A[i];
            }
        if (min<=n)
        {
            n-=min;
            count++;
            A[j]=1000;
        }
    }
    cout << count;
    return 1;
}

```

Ниже представлена программа на языке Python.

```

n,k=map(int,input().split())
a=list(map(int,input().split()))
min=1
count=0
while(min<1000):

```

```

min=1000
for i in range (0,k):
    if a[i]<=n and a[i]<min:
        j=i
        min=a[i]
if min<=n:
    n-=min
    a[j]=1000
count+=1
print(count)

```

Возможны и другие решения, например, можно упорядочить массив A по возрастанию, затем накапливать сумму первых элементов, пока она не превысит число n , зачем вывести максимальное число слагаемых, при котором сумма еще меньше n . По вычислительной сложности эти решения сравнимы. Ниже представлена программа на языке Python, реализующая эту идею.

```

n,k=map(int,input().split())
a=list(map(int,input().split()))
a=sorted(a)
print(a)
count=0
for x in a:
    n-=x
    if n<0:
        break
    else:
        count+=1
print(count)

```

Задача 2.2. На семинаре по новым методам газодобычи имеется конференц-зал с прекрасным современным оборудованием, в котором несколько участников хотят представить свои доклады. Для составления расписания участники подали заявки вида $[x,y)$ – время начала и окончания доклада. Доклад может начаться в момент окончания другого доклада, без перерыва. Предложите алгоритм для составления расписания докладов, при котором будет выполнено максимальное количество заявок. Напишите программу, реализующую данный алгоритм.

Входные данные

В первой строке вводится натуральное число N , не более 1000 – общее количество заявок. Затем вводится N строк с описаниями заявок - по два числа в каждой строке, x и y , $x < y$. Время начала и окончания доклада – натуральное число, не превышает 1440 (в минутах с начала суток).

Выходные данные

Выведите одно число – максимальное количество заявок, которые можно выполнить.

Пример

Входные данные

4

1 6

2 4

4 8

10 11

Выходные данные

3

Примечание. Можно выполнить вторую, третью и четвертую заявки.

Решение задачи 2.2. Данная задача широко известна как «задача о заявках» и также может быть решена жадным алгоритмом; выше, в решении задачи 2.1., даны ссылки на ее разбор. Решение основывается на следующем факте: чем раньше закончится текущий доклад, тем больше времени останется на другие доклады, и тем больше докладов можно успеть прочитать. Таким образом, каждый раз при выборе нового доклада мы будем искать доклад, который начинается не раньше, чем закончится текущий, а заканчивается как можно раньше. Таким образом, алгоритм может быть таким:

1. Записываем количество заявок в переменную n .
2. Сохраняем данные по времени обслуживания в массивы X и Y ; таким образом, интервал $[X[i], Y[y])$ характеризует заявку с номером i .
3. Обнуляем счетчик $count$ и текущее время t .

4. Находим минимальный элемент $Y[i]$ из тех элементов массива, для которых $X[i] \geq t$, запоминаем его в переменную min . Если такого элемента нет, выводим значение $count$ и заканчиваем.

5. Если такой элемент найден, увеличиваем $count$ на единицу, полагаем текущее время $t = min$ (окончание выбранного доклада).

6. Идем на шаг 4.

Далее представлена программа на языке Pascal.

```
Program Problem_2_2;
var n,t,count,min,i:integer;
X,Y:array[1..50] of integer;
begin
  readln(n);
  for i:=1 to n do
    read(X[i],Y[i]);
  t:=0;
  min:=0;
  count:=0;
  while(min<1500) do
  begin
    min:=1500;
    for i:=1 to n do
      if((X[i]>=t) and (Y[i]<min)) then
        min:=Y[i];
    if (min<1500) then
      begin
        t:=min;
        count:=count+1;
      end;
    end;
  writeln (count);
end.
```

Ниже приведена программа на языке C++.

```
#include<iostream>
using namespace std;

int main()
{
  int n,t,count,min,i,X[1000],Y[1000];
  cin >> n;
  for(i=0;i<n;i++)
    cin >> X[i] >> Y[i];
  t=0;
  min=0;
  count=0;
  while (min<1500)
  {
```

```

        min=1500;
        for(i=0;i<n;i++)
            if((X[i]>=t) && (Y[i]<min))
                min=Y[i];
        if (min<1500)
        {
            t=min;
            count++;
        }
    }
    cout << count;
    return 1;
}

```

Ниже приведена программа на языке Python.

```

N=int(input())
a=[];otvet=[];
for i in range(0,N):
    a.append(list(map(int,input().split())))
t=0
count=0
min=0
while min<1500:
    min=1500
    for i in range(0,N):
        if (a[i][0]>=t) and (a[i][1]<min):
            min=a[i][1]
    if min<1500:
        t=min
        count+=1
print(count)

```

Возможны и другие решения, например, можно упорядочить массив Y по возрастанию, не забывая параллельно преобразовывать массив X , затем просмотреть массивы один раз с начала до конца, выбирая подходящие заявки: для выбранных заявок $[X[j], Y[j]]$, $j = 1, \dots, count$ должно выполняться условие $Y[j] \leq X[j + 1]$. Это означает, что время начала следующей заявки не меньше, чем время окончания предыдущей. По вычислительной сложности эти решения сравнимы.

Ниже приведена программа на языке C++, реализующая эту идею. После ввода данных массив Y упорядочивается по возрастанию методом пузырька, параллельно соответственно меняется массив X . После этого текущее время t полагается равным нулю. Массив X просматривается с начала, если находится подходящая заявка, начало которой не превосходит текущего времени, ($X[i] \geq t$), она добавляется в список: увеличивается счетчик $count$, и текущее время приравнивается ко времени завершения заявки, т.е., полагается $t = Y[i]$. В конце выводится значение $count$.

```

#include<iostream>
using namespace std;

int main()
{
int n,t,c,count,i,j,X[1000],Y[1000];
cin >> n;
for(i=0;i<n;i++)
cin>> X[i] >> Y[i];
for(i=1;i<n;i++)
    for(j=0;j<n-i;j++)
        if(Y[j]>Y[j+1])
            {
                c=Y[j];
                Y[j]=Y[j+1];
                Y[j+1]=c;
                c=X[j];
                X[j]=X[j+1];
                X[j+1]=c;
            }
t=0;
count=0;
for(i=0;i<n;i++)
    {
        if(X[i]>=t)
            {
                t=Y[i];
                count++;
            }
    }
cout<<count;
return 1;
}

```

Ниже приведена реализующая эту идею программа на C++, представленная одним из участников олимпиады. Программа использует встроенную функцию сортировки для вектора пар.

```

#include<iostream>
#include<algorithm>
#include<vector>

using namespace std;

int main()
{
    int n;
    cin >> n;
    vector<pair<int,int>> arr(n);
    for(int i=0;i<n;i++)

```

```
        cin >> arr[i].second >> arr[i].first;
    sort(arr.begin(),arr.end());
    int ans=0, r=0;
    for(int i=0;i<n;i++)
        if (r<=arr[i].second)
        {
            ans++;
            r=arr[i].first;
        }
    cout << ans << endl;
}
```

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.
- ✓ Дейнерис смотрит сериал «Как я встретил вашу маму».
- ✓ Палатка Дейнерис 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 разные два дня, между которыми прошло не более пяти дней;

2) SUM делится на 14;

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.

Программа должна напечатать отчет по следующей форме:

Контрольное значение:

Отчет сформирован. (или — Отчет не сформирован)

Кратко опишите используемый алгоритм решения.

На вход программе в первой строке подается количество дней N ($5 < N < 5000$), далее в N строках подаются объемы добычи газа (натуральные числа, не превышающие 500). В последней строке записано контрольное значение.

Решение задачи 4.

Сразу отметим, что для любого значения измерения B в паре с ним имеет смысл рассматривать лишь те измерения A , которые были сделаны не более чем за пять дней до B . Поэтому для хранения данных достаточно массива A размерности 5, в который записываются измерения за последние 5 дней. Переменная sum инициализируется значением -1 . В первом цикле считываются измерения за первые пять дней, и для них рассматриваются все возможные пары измерений, полученных в разные дни, т.е., измерения сохраняются в массив A , и рассматриваются все пары различных элементов массива $A[i] + A[j]$. Если $A[i]$ и $A[j]$ удовлетворяют приведенным условиям, и $A[i] + A[j] > sum$, то значение sum обновляется и становится равным $A[i] + A[j]$. Во втором цикле считывается очередное измерение x , рассматривается в паре со всеми предыдущими измерениями за последние 5 дней, т.е., со всеми элементами массива A . Если сумма x и $A[j]$ удовлетворяют условиям задачи, и $x + A[j] > sum$, то значение sum обновляется и становится равным $x + A[j]$. После этого происходит сдвиг влево элементов массива, и последнему элементу массива присваивается значение x . В конце считывается контрольное значение и сравнивается с переменной sum .

Возможны и другие решения, в частности, вместо сдвига массива можно использовать деление с остатком. 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.

Ниже приведена программа на языке Pascal.

```
Program Problem_4
var x, y, z, i, j, n, sum: integer;
    A: array[1..5] of integer;
begin
    sum := -1;
    readln(n);
    readln(A[1]);
    for i := 2 to 5 do
    begin
        readln(A[i]);
        for j := 1 to i-1 do
        begin
            y := A[i]*A[j];
            z := A[i]+A[j];
            if ((y mod 6 <> 0) and (z mod 14 = 0) and (z > sum)) then
                sum := z;
        end;
    end;
    for i := 6 to n do
    begin
        readln(x);
        for j := 1 to 5 do
        begin
            y := x*A[j];
            z := x+A[j];
            if ((y mod 6 <> 0) and (z mod 14 = 0) and (z > sum)) then
                sum := z;
        end;
        for j := 1 to 4 do
            A[j] := A[j+1];
        A[5] := x;
    end;
    writeln ('Контрольное значение: ', sum);
    readln(x);
    if ((x = sum) and (sum > -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 x,y,z,i,j,n,sum,A[5];
    sum=-1;
    cin >> n >> A[0];
    for(i=1;i<5;i++)
    {
        cin>> A[i];
        for(j=0;j<i;j++)
        {
            y=A[i]*A[j];
            z=A[i]+A[j];
I            f((y%6!=0) && (z%14==0) && (z>sum))
                sum=z;
        }
    }
    for(i=5;i<n;i++)
    {
        cin>> x;
        for(j=0;j<5;j++)
        {
            y=x*A[j];
            z=x+A[j];
            if((y%6!=0) && (z%14==0) && (z>sum))
                sum=z;
        }
        for(j=0;j<4;j++)
            A[j]=A[j+1];
        A[4]=x;
    }
    cout << "Контрольное значение: " << sum << endl;
    cin >> x;
    if ((x==sum) && (sum>-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		

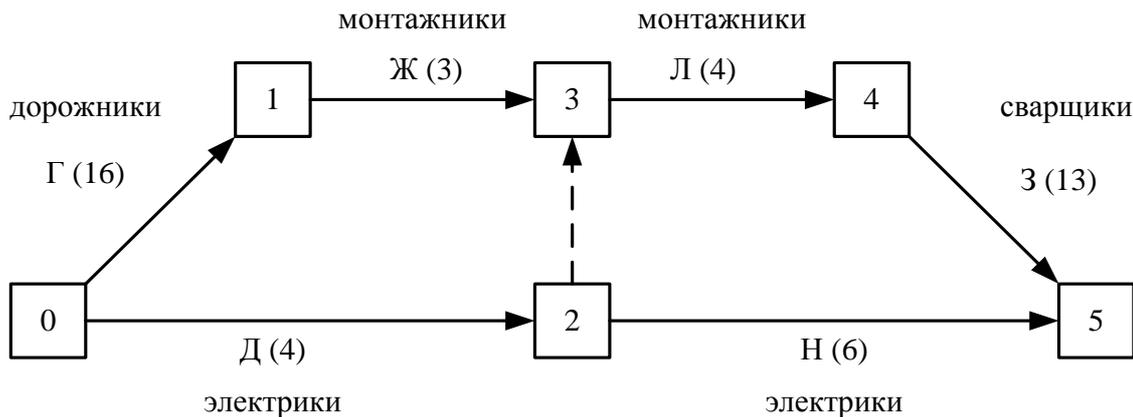
Определите минимальный срок, за который можно закончить подготовительные работы на месторождении Дальнем. Сколько сотрудников каждой профессии при этом понадобится?

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 выполняют 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 пути	Дорожники	182	16	12
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.	Подводэлектроэнергии	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. Жадные алгоритмы

Задача 6. Для проведения геолого-технических мероприятий на фонде скважин предприятия Дотракия необходимо произвести некоторые измерения. Значения этих измерений передаются в зашифрованном виде. Конкуренты с предприятия Дальнее перехватили часть незашифрованных и зашифрованных данных и хотят разгадать шифр, а также отправить свое сообщение в зашифрованном виде.

Известно, что незашифрованные данные содержат количество цифр, кратное трем, и для некоторых данных известны шифрованные сообщения. Они приведены в таблице.

Незашифрованные данные	Зашифрованное сообщение
231427	25434537
320474	31204771
344061	32310303
726543	75345226

Также удалось узнать, что для шифрования используется двоичная система счисления. Помогите предприятию Дальнему разгадать шифр.

Зашифруйте данные: 653412.

Ответ обоснуйте.

Решение задачи 6. Прежде всего заметим, что в сообщениях приводятся цифры от 0 до 7, т.е., все сообщения можно рассматривать как последовательности цифр в восьмеричной системе счисления. Для перевода сообщений в двоичную систему счисления каждая цифра заменяется ее двоичным кодом длины три. Переведем сообщения в двоичную систему счисления.

Незашифрованные данные		Зашифрованное сообщение	
Исходное	Двоичный код	Исходное	Двоичный код
231427	010011001100010111	25434537	010101100011100101011111
320474	011010000100111100	31204771	011001010000100111111001
344061	011100100000110001	32310303	011010011001000011000011
726543	111010110101100011	75345226	111101011100101010010110

После шифрования 6 цифр превращаются в 8, т.е., при шифровании к исходным 18 битам добавляется 6 бит. Заметим, что первая цифра не меняется, что позволяет предположить, что бит добавляется после первой цифры. В исходном сообщении 6 цифр, добавлено 6 бит, предположим, бит добавляется после каждой цифры. Проверим это предположение: выпишем исходный двоичный код и полученный двоичный код друг под другом, в исходном коде после каждых трех бит вставим пробел.

Исходное	010 011 001 100 010 111
Зашифрованное	010 1 011 1 000 1 1100 1 010 1 11 1 1
Исходное	011 010 000 100 111 100
Зашифрованное	011 0 010 1 000 0 100 1 111 1 100 1
Исходное	011 100 100 000 110 001
Зашифрованное	011 0 100 1 100 1 000 0 110 0 00 1 1
Исходное	111 010 110 101 100 011
Зашифрованное	111 1 010 1 110 0 101 0 100 1 01 1 0

Догадка оказалась верна – последовательности из трех бит в исходном и в зашифрованном сообщении совпадают, т.е., при шифровании после трех бит добавляется четвертый (выделен жирным шрифтом). В некоторых случаях он равен 0, в некоторых – 1. Попробуем найти закономерность. Она не связана с позицией бита, т.к. в первом случае добавляются биты 101111, во втором – 010111, и т.д. Значит, они зависят от цифр числа. Бит 0 добавляется после последовательностей 000, 011, 101, 110; бит 1 – после последовательностей 001, 010, 100, 111. Теперь ясно, что четвертый бит представляет собой проверку на четность числа единиц в предыдущих трех битах: число единиц в предыдущих трех битах делится на 2 без остатка, четвертый бит равен 0, иначе он равен 1. Шифр разгадан. Порядок шифрования такой:

1. Переводим очередную цифру в трехзначный двоичный код.
2. Если число единиц в коде четно, добавляем 0, иначе добавляем 1.
3. Если не все цифры зашифрованы, идем на шаг 1.

Зашифруем теперь число 653412, результат представим в таблице.

Число	6	5	3	4	1	2		
Двоичный код	110	101	011	100	001	010		
Результат шифрования в двоичном коде	110010100110100100110101							
Триады	110	010	100	110	100	100	110	101
Перевод в восьмеричную систему	6	2	4	6	4	4	6	5

Таким образом, зашифрованное сообщение 62464465.

В остальных вариантах алгоритм шифрования может отличаться, но строится на том же принципе – добавление бита после трех бит. Попробуйте разгадать шифры самостоятельно.

Возможно, вы нашли другой алгоритм шифрования, приводящий к тому же результату.

Ответ. Переводим очередную цифру в трехзначный двоичный код. Если число единиц в коде четно, добавляем 0, иначе добавляем 1. Зашифрованное сообщение 62464465.

7. Комбинаторика

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

- 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 3 существует больше, чем длины 2, рассмотрим сначала все возможные последовательности длины 3, удовлетворяющие условиям. Последовательность не начинается с сигнала зеленого цвета, поэтому первый сигнал можно выбрать двумя способами: красный и белый. Так как все сигналы различны, второй сигнал затем можно выбрать двумя способами: зеленый и белый (для первого красного), зеленый и красный (для первого белого). Для третьего сигнала остается один вариант в каждом случае. Итого для трех ракет можно составить $2 \times 2 \times 1 = 4$ сигнала.

Рассмотрим теперь последовательности длины 2. Они не должны совпадать с началом последовательности длины 3, поэтому можно начать с зеленого сигнала, тогда второй будет либо красным, либо белым (два варианта). Кроме этого, возможна последовательность из двух белых сигналов подряд.

Итого, можно закодировать семь сообщений.

КЗБ

КБЗ

БЗК

БКЗ

ЗБ

ЗК

ББ

Задачи в остальных вариантах решаются аналогично.

Ответ. 7 сообщений.

8. Кодирование и декодирование

Задача 8. Наша служба безопасности заметила, что кто-то передает информацию о разрабатываемых месторождениях. И вот им наконец удалось перехватить сообщение и зашифрованный файл. Похоже, что это сообщение и есть ключ к файлу. Сообщение содержало следующую информацию:

$X\bar{X}Y\bar{Y}Y\vee\bar{X}$	$\bar{X}YX\bar{X}\bar{Y}$	$\bar{X}XY\bar{Y}\bar{X}$	$X\bar{X}Y\bar{X}\bar{Y}$	$\bar{Y}Y\bar{X}XY\wedge X$
--------------------------------	---------------------------	---------------------------	---------------------------	-----------------------------

Двумя днями ранее мы перехватили сообщение $X=1, Y=0$. Так же в сообщении сказано, что шифром является слово. Расшифруй данное закодированное слово.

Решение задачи 8: Сперва необходимо подставить значения X и Y . Будет получена таблица с 0 и 1. Они уже сгруппированы по 5 знаков. Получается двоичный код пятого разряда, мы знаем, что тут зашифровано. Если сопоставить кодам буквы русского алфавита, в соответствии с их порядковыми номерами, то выйдет слово:

1	0	0	1	0	т
0	0	1	0	1	е
0	1	0	1	0	к
1	0	0	0	1	с
1	0	0	1	0	т

Ответ: ТЕКСТ

Также можно пронумеровать таблицу истинности и затем подставить буквы в соответствии номеру в алфавите. Важно, что отсчет начинается с 0, а не с единицы, так как $00000 = 0$. Также в русском алфавите буква “е” приравнивается “ё”. Если “ё” будет учтена как отдельная буква, то все буквы начиная с “е” сместятся на одну позицию. Так, к примеру, вместо слова ТЕКСТ получится СЕЙРС.

9. Количество информации

Задача 9. Два датчика (А и Б) передают поочередно статистическую информацию о состоянии буровой скважины, сначала данные передаёт датчик А, затем Б, после этого следует пауза в 180 секунд. Из-за большого количества помех часть данных искажается и считается потерянной. При передаче от А происходят потери 10 %, при передаче от Б потери равны 20 %. Скорость передачи от обоих датчиков 10 бит/с. Размер передаваемой информации в обоих случаях 10 байт. Первая передача происходит на 0 секунде. Рассчитать количество общей потерянной информации за 11 мин и представить в виде Бит.

Решение задачи 9:

Вычисляем по известным данным неизвестные. Знаем, что А передает 10 байт, потери 10% = 1 байту, потери Б = 2 байта.

Сначала требуется рассчитать количество передач. Скорость и размер файлов одинаков, скоростью 10бит/с. 10 байт $\times 8 = 80$ бит. У каждого уйдет по 8 секунд плюс после передачи задержка 180 секунд, т.е. один полный цикл равен 196 секундам.

11 минут это 660 секунд. $660 / 196 = 3,37$. Т.е. произойдет 3 полных передачи, включая задержку, следовательно, следующая передача успеет произойти, а последующая за ней задержка нет.

Зная, что пройти успеют четыре передачи, находим общие потери за четыре передачи. Для А потери 1 байт или 8 бит, для Б 2 байта или 16 бит.

$$8 \times 4 + 16 \times 4 = 32 + 64 = 96 \text{ бит}$$

Ответ: 96 бит

10. Системы счисления

Задача 10. Газ добывается на 5 станциях и затем подается в общий трубопровод для дальнейшей транспортировки. Все станции строились в разное время и используют разные системы счисления для подсчета объема добываемого газа.

Первая и вторая буровые ведут подсчет в пятеричной системе счисления. Первая добывает 10030000 м^3 , при этом вторая добывает в десять раз меньше при мощности работы в 40%.

Третья станция также использует пятеричную систему счисления и добывает половину от суммы первой и второй станций при мощности в 80%.

Четвертая станция добывает в два раза больше пятой, а Пятая добывает $186A0 \text{ м}^3$ и использует шестнадцатеричную систему счисления.

После поступления на Основной узел весь газ отправляется дальше.

Найти максимальный объем поставляемого газа при работе всех станций на полную мощность. Напишите ответ в виде числа в десятичной системе счисления.

Примечание: Если система счисления не указана, значит, число в десятичной системе счисления

Решение задачи 10

Первая добывает $1003000000(5)$, при переводе в десятичную систему это 80000.

Вторая добывает в десять раз меньше при мощности 40%, это значит 8000. При 100% мощности она добывает $\frac{8000 \times 100}{40} = 20000$.

Третья добывает половину суммы первой и второй, это $(80000+8000)/2=88000/2$ при мощности 80% , это 44000. Значит при 100% мощности она добывает $\frac{44000 \times 100}{80} = 55000$.

Пятая добывает $186A0$ это 100 000 в десятичной системе счисления.

Четвертая добывает $100\ 000 \times 2 = 200\ 000$

Складываем все числа: $80\ 000 + 20\ 000 + 55000 + 100\ 000 + 200\ 000 = 455\ 000$

Ответ: $455\ 000 \text{ м}^3$.