1. Задача: «*Кенгуру – 100 баллов*». (*100 баллов*).

Суперкенгуру может прыгать по прямой вправо и влево и совершать гигантские прыжки. Длина его первого прыжка составляет 1 м, второго — 2 м, третьего — 4 м и так далее (длина каждого прыжка всегда в два раза больше, чем предыдущего). Через какое минимальное количество прыжков суперкенгуру окажется на расстоянии **d** метров от исходной точки О?

Формат ввода:

d – расстояние **d** (натуральное число, 0 < **d <** 2147483647 = 2^{31}–1) метров от исходной точки О.

Формат вывода:

Минимальное количество прыжков для попадания в точку на расстоянии **d** или сообщить, что это невозможно.

Примеры:

Расстояние 16 Невозможно

Расстояние 17 Минимальное число шагов 5

Примечание. Имеется в виду последовательность прыжков –1–2–4+8+16, дающая расстояние 17.

Задача 2: *Набор букв. (100 баллов).*

На карточках написан некий набор букв, причем на каждой карточке – только одна буква. Вывести любую пару слов, которую можно составить из набора. Если 2 слова составить невозможно, то вывести «NO».

Набор букв (без пробелов) расположен в первой строке, в каждой из следующих пяти – слово. Все строки имеют длину не более 100 символов.

Пример.

Ввод	**Вывод**
МАРТРИОТРА {набор букв} МИР{1-е слово} КИТ {2-е слово} КРЫМ {3-е слово} КОТ {4-е слово} ТОРТ {5-е слово}	МИР {1-е составленное слово} ТОРТ {2-е составленное слово}

Задача 3. Рождественский пирог. (50 баллов).

Пекарь считает, что для получения рождественского пирога идеальной симметричной формы его нужно вынимать из духовки в тот момент, когда часы показывают «палиндромное» время, которое читается одинаково слева-направо и справа-налево. Напишите программу, которая определяет по времени установки пирога в духовку время, когда будет подходящее время для его извлечения.

Входные данные: время установки пирога в духовку в формате
HH:MM (00 ≤ HH ≤ 23, 00 ≤ MM ≤ 59)

Выходные данные: ближайшее «палиндромное» время в формате HH:MM.

Ввод	**Вывод**
00 00	01:10
12 34	13:31
23 59	00:00

Задача 4. Архиватор. (Баллов 50).

Однажды программист Вася решил послать своему другу письмо. Он подробно рассказал в нем, как у него дела. И, в конце концов, вышло большое послание. Но почта не позволяет пересылать такие большие письма. После долгих раздумий Вася решил его сжать и для этого придумал оригинальный метод. Если подряд идут несколько одинаковых символов, а точнее три и более, то он их заменяет на их количество и символ, который повторяется. Но Вася был плохим программистом... Помогите ему сжать письмо.

Входные данные: s
Выходные данные: K
Где: S-письмо не более 255 символов, K-сжатое письмо

Пример ввода: GGHHHHGDDDDHHHKKKKKJJJ

Пример вывода: GG4HG4D3H5K3J

Ответы и тесты:

Задача 1: «*Кенгуру – 100 баллов*».

Каждый тест по 20 баллов

Тест	Исходные данные	Результат	Кол-во баллов	Примечание (как образован и почему не меньше)
1	18	Невозможно	10	
2	19	5	20	1–2–4+8+16 = 19 1+2+4+8 = 15 < 19
3	255	8	20	1+2+4+8+16+32+64+128 = 255
4	111111	17	25	1+2–4–8–16–32–64–128+256–512–1024+2048+4096– –8192+16384+32768+65536 = 111111 1+2+4+8+16+32+64+128+256+512+1024+2048+4096 ++8192+16384+32768 = 65535 < 111111
5	2147483645	31	25	–1+2+4+8+16+32+64+128+256+512+1024+2048 +4096+8192+16384+32768+65536+131072+262144+ +524288+1048576+2097152+4194304+8388608+ +16777216+33664432+67108864+134217728+ +268435456+536870912+1073741824 = 2147483645 +1+2+4+8+16+32+64+128+256+512+1024+2048 +4096+8192+16384+32768+65536+131072+262144+ +524288+1048576+2097152+4194304+8388608+ +16777216+33664432+67108864+134217728+ +268435456+536870912 = 1073741823 < 2147483645

Решение «***Кенгуру – 100 баллов***»

Ясно, что кенгуру не может попасть в точку на четном расстоянии от O. Для нечетных d решение заключается в суммировании ряда $1+2+4+\ldots+2^i+\ldots$ с подсчетом числа сделанных прыжков до тех пор, пока сумма не станет равной или больше d. Найденное число прыжков и будет ответом. В самом деле, если сумма равна d, то кенгуру прыгает направо нужное число раз. А вот если больше?

Тоже верно. Для обоснования рассмотрим разность суммы и d. Ее можно разложить в по степеням 2. Если в этом разложении есть 2^i, то просто в сумме поменяем знак плюс на

минус у слагаемого 2^{i-1}. Это приведет к уменьшению разности между суммой и d на 2^i. И так с каждой компонентой разложения разности.

Ниже приведено решение и еще одно решение с выводом последовательности прыжков (минус – влево, плюс – вправо).

```
program task3;
  var
    a,r,s,st:longint;
begin
  writeln('Введите число ');
  readln(a);
  if a mod 2=0 then
      writeln('Невозможно')
    else
      begin
        r:=0;
        st:=1;
        s:=0;
        while s<a do
          begin
            r:=r+1;
            s:=s+st;
            if s<a then st:=st*2;
          end;
        writeln('Минимальное число шагов  ',r);
      end;
  readln;
end.
```

Решение с последовательностью прыжков.

```
program task3d;
  var
    a,r,s,st,i:longint;
    b:array [1..31] of longint;
begin
  writeln('Введите число ');
  readln(a);
  if a mod 2=0 then
      writeln('Невозможно')
    else
      begin
        r:=0;
        st:=1;
        s:=0;
        while s<a do
          begin
            r:=r+1;
            s:=s+st;
            if s<a then st:=st*2;
          end;
        writeln('Минимальное число шагов  ',r);
      end;
  s:=(s-a) div 2;
  st:=1;
  for i:=1 to r do
    begin
      if (s mod 2=0) then
```

```
        write('+',st)
      else
        write('-',st);
    if i<r then st:=st*2;
    s:=s div 2;
   end;
  readln;
end.
```

Решение: ***Набор букв. (100 баллов).***

Организуем перебор всех возможных пар слов из пяти заданных (циклы со счетчиками *i* и *j* в процедуре *resh).* Слова текущей пары *a[i]* и *a\j]* объединяем в одну строку *(sum2slov).* Далее организуем еще 2 цикла (со счетчиками *il* и *jl),* чтобы проверить наличие символов строки *sum2slov* в исходном наборе букв (переменная *s).* Если найдена одна и та же буква в строках *sum2slov* и *s,* то в дополнительном массиве *Ъ* ставится метка (число **1)** в позиции, соответствующей индексу буквы в строке *s.* Это сигнализирует о том, что буква уже использована и при повторных обращениях выбрана не будет. Количество совпадений букв в рассматриваемой паре слов *(sum2slov)* и в исходном наборе букв *(s)* регистрируем в переменной *kol.* Если по истечении циклов сравнения всех символов строк *sum2slov* и 5 значение переменной *kol* равно длине строки *sum2slov,* то найдена соответствующая пара слов. В этом случае производится вывод результатов в требуемом формате, а программа заканчивает работу.

Программа.

```
Program z3_3;
Var        i: integer;
s:string;
a:array[1..5]  of string[100]; b:array[1..100]  of byte;
Procedure resh;
Var        i,j,t,i1,j1,kol:integer;
sum2slov:string;
Begin
for i:=1 to 4 do        {Циклы по i и j для выбора 2-х слов}
 for j:=i+1 to 5 do Begin
for t:=1 to 100 do b[t] :=0; {Обнуление массива b}
kol:=0;
sum2slov:=a[i]+a[j];
if length(sum2slov)<=length(s)   then
for i1:=1 to length (sum2slov)  do
begin
for j1:=1 to length(s)  do
if  (b[j1]=0)  and  (s[j1]=sum2slov[i1])
then       {Найден символ из sum2slov в s}
Begin
b[j1]:=1;      kol:=kol+1; break; End;
if kol<i1 then break; {Есть ненайденные символы в s}
end;
if kol=length(sum2slov) then          {Найдена пара слов, удовлетворяющих условию}
```

```
Begin writeln(a[i]) ;    writeln(a[j]); halt;
End;
End;
End;
```

№ теста	Входные данные	Выходные данные (возможны и другие варианты вывода слов)	Кол-во баллов
1	АААА…А {10 БУКВ А} УХА МИР МАМА ПАПА РИМ	NO	10
2	МГОТАПМИАПААДНР ТОМ КИТ РОТ МАТ ПАР	ТОМ ПАР	15
3	МГОТАПМИАПААДНР КИТ ФУТ СОК ПЛАН ПАР	NO	15
4	МГОТАПМИАПААДНР НЕТ ДЫМ ДЕНЬ ПИРАМИДА ДОМ	NO	15
5	МГОТАПМИАПААДНР ДОМ НЕТ КИТ КОТ ПАПА	ДОМ ПАПА	15
6	МГОТАПМИАПААДНР НЕТ КОТ ДА БЫК РАМА	РАМА ДА	15
7	МГОТАПМИАПААДНР РИМ ТИР БОР КИТ МАМА	ТИР МАМА	15

Задача 3. Рождественский пирог (50 баллов)

№	Ввод	Вывод	Баллы
1	00 00 может быть 0 0	01:10	10
2	12 34	13:31	10
3	23 59	00:00	10
4	05 14	05:50	10
5	12 00	12 21	10

Программа:

```
var h,m:integer;
begin
read(h,m);
if   (h div 10 =m mod 10)and(h mod 10=m div 10) then m:=m+1;
while (h div 10 <>m mod 10)or(h mod 10<>m div 10) do
if m+1=60 then
   begin
       m:=0;
       h:=h+1;
       if h=24 then h:=0;
       end
   else m:=m+1;
   if h<10 then write(0,h,':') else write(h,':');
    if m<10 then write(0,m) else write(m);
end.
```

Задача 4. Архиватор

Программа:

```
program stroki;
var
  s     : string;
  i,d,k : longint;
begin
  readln(s);
  s:=s+'й';
  d:=length(s);
```

```
  k:=1;
  for i:=1 to d-1 do
    if s[i]=s[i+1]
      then k:=k+1
      else begin
             if k=1 then write(s[i]);
             if k=2 then write(s[i],s[i]);
             if k>2 then write(k,s[i]);
             k:=1;
           end;
end.
```

№	Ввод	Вывод	Баллы
1	GGHHHH	GG4H	10
2	DDDDDD	6D	10
3	HHHKKKKKJJJ	3H5K3J	10
4	GHGDHKJ	GHGDHKJ	10
5	HGDDDDHHHKKKKK	HG4D3H5K	10

Задания для первого этапа Республиканской олимпиады по учебному предмету «Информатика» в 2022/2023 учебном году

Требования к организации и проведению первого этапа Республиканской олимпиады по учебному предмету «Информатика»:

Задания олимпиады выполняются на компьютере на базе учреждений образования. Количество задач – четыре.

В олимпиаде могут принимать участие учащиеся 6-11 классов учреждений образования. Участник олимпиады имеет право выполнять задачи на различных алгоритмических языках или в различных средах программирования.

Решением задачи является программа на одном из допустимых языков программирования.

Продолжительность первого этапа олимпиады – 2 часа с момента получения задания участниками.

Результаты работы программы проверяет жюри учебного заведения, которое вправе составить набор тестов для проверки каждого задания (или воспользоваться предлагаемым набором тестов). Каждый тест из этого набора оценивается своим количеством баллов. Если тест пройден, участник олимпиады получает количество баллов, соответствующих этому тесту. Результат выступления участника – сумма баллов, полученных за все пройденные тесты.