Олимпиада «Ломоносов», информатика, 2011 год, вариант 1.

Задача 1.

Ответ. 0,5 мибит/с.

Решение. Пусть v — искомая скорость в мибит/с, x — объем MP3-файла в представлении Base64 в мебибитах, тогда $\frac{3x}{4}$ — исходный размер MP3-файла; $\frac{x}{2v}$ — время доставки первой половины файла по электронной почте; $\frac{x}{2(v+2)}$ — время доставки второй половины файла по электронной форме; $\frac{6x}{4v+3}$ — удвоенное время скачивания исходного файла с Викиликс. Получаем уравнение $\frac{x}{2v} + \frac{x}{2(v+2)} = \frac{6x}{4v+3}$. Так как $x > 0$, $v > 0$, получаем уравнение $2v^2 + 5v - 3 = 0$.

Задача 2.

Ответ: 8 или 9 бит (416 различных сообщений).

Решение. Всего различных размещений горшков: 440. Пять горшков из восьми могут быть выбраны по одной из следующих схем:

1. $3 + 2$: три горшка одного типа (Кактус), два горшка другого типа; таких размещений $2 \cdot \frac{5!}{3! \cdot 2!} = 20$, где 2 - число способов выбрать тип горшка (один из двух);
2. $3 + 1 + 1$: три горшка одного типа, два горшка разных типов; таких размещений $3 \cdot \frac{5!}{3!} = 60$, где 3 - число способов выбрать два типа горшков (два из трех);
3. $2 + 2 + 1$: два горшка одного типа, два горшка другого типа, один горшок третьего типа; таких размещений $3 \cdot 2 \cdot \frac{5!}{2! \cdot 2!} = 180$; 3 — число способов выбрать два типа горшков из трех, которых берется по два горшка, 2 — число способов выбрать один тип горшков из двух оставшихся;
4. $2 + 1 + 1 + 1$; таких размещений $3 \cdot \frac{5!}{2!} = 180$.

Размещений горшков с тремя подряд — 24: 6 по схеме «3+2» и 18 по схеме «3+1+1».

Задача 3.

Ответ: красный и желтый провода не взрывают бомбу, зеленый провод взрывает бомбу.

Решение. Обозначим высказывания: A — красный провод взрывает бомбу, B — желтый провод взрывает бомбу, C — зеленый провод взрывает бомбу. Тогда на записках написаны формулы $A \vee B \wedge \neg C$, $\neg A \wedge B$, $A \wedge B \wedge C$.

Предположим, что все записки одновременно истинны, то есть $(A \vee B \wedge \neg C) \wedge (\neg A \wedge B) \wedge (A \wedge B \wedge C) = 1$. Это невозможно.

Предположим, что все записки одновременно ложны, то есть $\neg(A \vee B \wedge \neg C) \wedge \neg(\neg A \wedge B) \wedge \neg(A \wedge B \wedge C) = 1$. Отсюда $\neg A \wedge (\neg B \vee C) \wedge (A \vee \neg B) \wedge (\neg A \vee \neg B \vee \neg C) = 1$. Отсюда $\neg A = 1$, $\neg B = 1$. Так как $A \vee B \vee C = 1$, то $C = 1$.

Задача 4.

Ответ: DCB6BCD.

Решение. В системе счисления с основанием 14 число делится на 13 тогда и только тогда, когда сумма его цифр делится на 13 (участники олимпиады должны это утверждение обосновывать). Поскольку каждую цифру можно использовать не более двух раз, палиндром, очевидно, будет иметь вид `DCBxBCD`, где на месте `x` стоит такая цифра, чтобы сумма цифр числа делилась на 13. Эта цифра — 6, так как $0 + -1 + -2 + x + -2 + -1 + 0 = 0$ (остатки от деления на 13 всех цифр числа).

Задача 5.

Ответ:

```
10 2 -3
10 6  5
10 1 -1
```

Решение: После выполнения операций по копированию формул получается следующая таблица:

```
=SUM(B$1:$C2) =SUM(C$1:$C2) C1
=SUM(B$1:$C3) =2*B$1+B1     C2
=SUM(B$1:$C4) =SUM(C$1:$C4) =2*C$1+C2
```

Выражая формулы в ячейках через C1 и C2 получаем таблицу:

```
=5*C1+5*C2  =C1+C2     C1
=10*C1+8*C2 =3*C1+3*C2 C2
=10*C1+8*C2 =3*C1+2*C2 =2*C1+C2
```

Таким образом, $10 \cdot C1 + 8 \cdot C2 = 10$, $3 \cdot C1 + 2 \cdot C2 = 1$, отсюда $C1 = -3$, $C2 = 5$.

Задача 6.

Ответ. Ответом является любая из следующих последовательностей:

```
1 2 6 5 4 3; 1 3 5 6 4 2; 1 3 6 4 5 2; 1 3 6 5 2 4; 1 4 3 6 5 2
1 4 5 3 6 2; 1 4 5 6 2 3; 1 4 6 2 5 3; 1 4 6 3 2 5; 1 5 2 6 4 3
1 5 3 4 6 2; 1 5 3 6 2 4; 1 5 4 2 6 3; 1 5 4 3 2 6; 1 5 6 2 3 4
1 6 2 4 5 3; 1 6 2 5 3 4; 1 6 3 2 5 4; 1 6 3 4 2 5; 1 6 4 2 3 5
2 1 5 6 4 3; 2 1 6 4 5 3; 2 1 6 5 3 4; 2 3 4 6 5 1; 2 3 5 4 6 1
2 3 5 6 1 4; 2 3 6 1 5 4; 2 3 6 4 1 5; 2 4 1 6 5 3; 2 4 3 5 6 1
2 4 3 6 1 5; 2 4 5 1 6 3; 2 4 5 3 1 6; 2 4 6 1 3 5; 2 5 1 4 6 3
2 5 1 6 3 4; 2 5 3 1 6 4; 2 5 3 4 1 6; 2 5 4 1 3 6; 2 6 1 3 5 4
2 6 1 4 3 5; 2 6 3 1 4 5; 3 1 4 6 5 2; 3 1 5 4 6 2; 3 1 5 6 2 4
3 1 6 2 5 4; 3 1 6 4 2 5; 3 2 1 6 5 4; 3 2 4 5 6 1; 3 2 4 6 1 5
3 2 5 1 6 4; 3 2 5 4 1 6; 3 2 6 1 4 5; 3 4 1 5 6 2; 3 4 1 6 2 5
3 4 2 1 6 5; 3 4 2 5 1 6; 3 4 5 1 2 6; 3 5 1 2 6 4; 3 5 1 4 2 6
3 5 2 1 4 6; 3 6 1 2 4 5; 4 1 2 6 5 3; 4 1 3 5 6 2; 4 1 3 6 2 5
4 1 5 2 6 3; 4 1 5 3 2 6; 4 1 6 2 3 5; 4 2 1 5 6 3; 4 2 1 6 3 5
4 2 3 1 6 5; 4 2 3 5 1 6; 4 2 5 1 3 6; 4 3 1 2 6 5; 4 3 1 5 2 6
4 3 2 1 5 6; 4 5 1 2 3 6; 5 1 2 4 6 3; 5 1 2 6 3 4; 5 1 3 2 6 4
5 1 3 4 2 6; 5 1 4 2 3 6; 5 2 1 3 6 4; 5 2 1 4 3 6; 5 2 3 1 4 6
5 3 1 2 4 6; 6 1 2 3 5 4; 6 1 2 4 3 5; 6 1 3 2 4 5; 6 2 1 3 4 5
```

Всего таких последовательностей 90 (из 720 всех возможных перестановок).

В качестве обоснования возможно моделирование работы алгоритма.

Задача 7.

Идея решения.

Данное решение работает за линейное время от длины массива. Все решения, работающие за большее время, в том числе использующие сортировку, считать неверными.

Предположим, что нужно разбить массив на две части так, чтобы в начале массива шли элементы, удовлетворяющие некоторому свойству, а в конце — все остальные. Эту задачу решает шаг разбиения массива алгоритма быстрой сортировки:

```
int a[N];
//...
i = 0; j = N - 1;
```

```
while (i < j) {
  while (i < j && is_prop(a[i])) ++i;
  while (i < j && !is_prop(a[j])) --j;
  swap(a[i++], a[j--]);
}
```

Дважды применяя алгоритм разбиения получаем следующую программу.

```
  i = 0; j = N - 1;
  while (i < j) {
    while (i < j && (a[i] % 7 == 5 || a[i] % 7 == 2)) ++i;
    while (i < j && (a[j] % 7 != 5 && a[j] % 7 != 2)) --j;
    if (i < j) swap(a[i++], a[j--]);
  }
  if (j < 0 || (a[j] % 7 == 5 || a[j] % 7 == 2)) ++j;
  i = 0;
  while (i < j) {
    while (i < j && a[i] % 7 == 5) ++i;
    while (i < j && a[j] % 7 != 5) --j;
    if (i < j) swap(a[i++], a[j--]);
  }
```

Задача 8.
Ответ.

```
I=S(I[1,1],I[1,1])
N3=S(N,I[3,3])
A=R(I,N3)
A3=S(A,I[1,3],I[3,3])
M=R(O,A3)
MULT=S(M,I[1,3],I[3,3])
```