

§2 Второй отборочный этап

Второй отборочный этап проводится в командном формате в сети интернет на платформе stepic.org.

Второй этап проходил с 6 ноября по 12 декабря. Блоки задач открывались участникам в онлайн-системе последовательно.

Расписание открытия блоков задач:

- Блок задач №1: с 6 ноября
- Блок задач №2: с 15 ноября
- Блок задач № 3: с 23 ноября

Задачи для 9-11 классов носят междисциплинарный характер и направлены на решение задач по обработке биосигналов человека – данная тема является одной из ключевых в Финальном задании. Решение каждой задачи дает определенное количество баллов.

В данном этапе можно получить суммарно от 0 до 125 баллов.

В задачах 1.1, 1.2, 1.3, 2.3, 3.1, 3.2 (задачи на программирование) баллы выставляются пропорционально пройденным тестам (например, если максимальное количество баллов за задачу 10, количество тестов - 2, решение участника прошло 1 тест, то участник получает 5 баллов за задачу). Количество попыток ввода решения в этих задачах не ограничено.

В задачах 2.1 и 2.2 (необходимо дать численный ответ) полный балл ставится за верный ответ, в противном случае – ноль баллов. Количество попыток в этих задачах – одна от команды (вторая и последующие попытки от команды не учитываются).

2.1 Задачи и решения

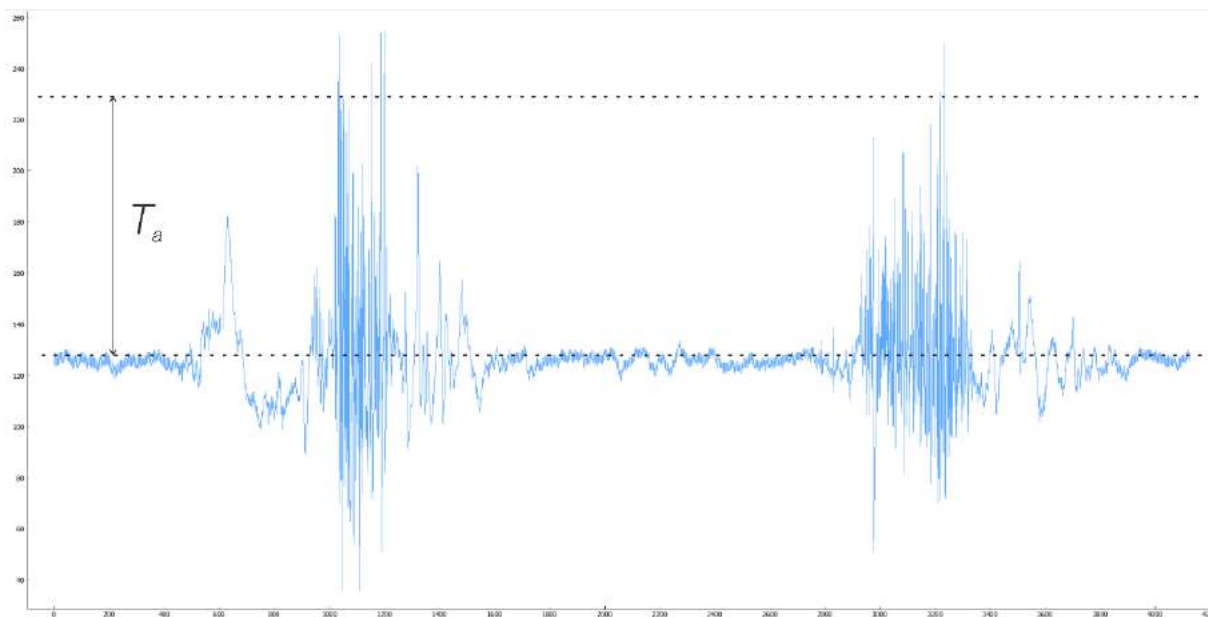
Блок задач №1

Задача 1.1 (10 баллов)

В данной задаче вам предлагается определить количество пучностей на записи сигнала электромиограммы (ЭМГ) — электрического сигнала, возникающего при напряжении мышцы человека. Данные, представленные в этой задаче, были сняты с мышцы руки. Сигнал на выходе представляет из себя последовательность целых чисел, разделенных запятой, принадлежащих диапазону от 00 до 255255. Средний уровень сигнала в состоянии покоя равен половине максимального значения. Пучностью будем считать близкое скопление амплитуд сигнала, превосходящих значение $T_a=100T_a=100$. По условию задачи, пучности не могут быть ближе, чем на расстоянии $N=1000N=1000$ точек между друг другом. Условимся считать за координату пучности точку с наибольшей амплитудой. Пример входного сигнала в формате csv (числа, разделенные запятой), доступен по ссылке: <http://bit.ly/2hxzRs3>

Приводим ниже его визуализацию:

Как вы можете убедиться, в представленном сигнале 2 пучности. Ваши решения будут тестироваться на более длительном сигнале, для проверки корректности их работы вы можете использовать данный образец.



Решение. Ниже приведено возможное решение на языке Python 3.

```
import sys
inp = sys.stdin.readline()
num = 0
points = list(map(int, inp.split(',')))
i = 0
while i < len(points):
    if (abs(points[i] - 127)) > 100:
        max = abs(points[i] - 127)
        n_max = i
        if (i + 1000 > len(points) - 1):
            end = len(points)
        else:
            end = i + 1000
        for j in range(i, end):
            if (abs(points[j] - 127) > max):
                max = abs(points[j] - 127)
                n_max = j
        i = n_max + 1000
        num += 1;
    else:
        i += 1
print(num)
```

Для проверки решений используется следующий код на языке Python (см. Приложение 1).

Задача 1.2 (15 баллов)

Как и в предыдущей задаче, входными данными в этой задаче являются сигналы ЭМГ с мышцы руки (сигнал на выходе представляет из себя последовательность целых чисел, разделенных запятой, принадлежащих диапазону от 0 до 255, средний уровень сигнала в состоянии покоя равен половине максимального значения).

Однако теперь вам предлагается проанализировать 2 сигнала, снятых одновременно. Необходимо узнать, в какие моменты времени мышцы рук были напряжены синхронно. Для этого мы введем кванты времени, содержащие 5 точек. Например, сигнал, состоящий из 100 точек, будет содержать 20 квантов времени.

Будем считать, что мышца напряжена в k -ый квант времени, если амплитуда сигнала хотя бы в одной точке этого кванта превосходит $T_a = 100$. Ответом на задачу будет являться строка

длиной в количество квантов времени сигналов, содержащая 1 на m -ом месте, если в m -ый квант времени мышцы были напряжены, и 0 в противном случае.

Гарантируется, что количество точек двух сигналов одинаково и кратно 5. Точки каждого сигнала разделены запятой, друг от друга сигналы отделены символом перевода строки.

Пример входных данных и ответ для данного примера доступен по ссылке:

<http://bit.ly/2hVWUge>

Решение. Ниже приведено возможное решение на языке Python 3.

```
import sys
import numpy as np
inp = sys.stdin.readline()
points1 = list(map(int, inp.split(',')))
inp = sys.stdin.readline()
points2 = list(map(int, inp.split(',')))
res1 = [0]*(int(len(points1)/5))
res2 = [0]*(int(len(points1)/5))
j = 0
i = 0
while i < len(points1):
    j = int(i / 5)
    if (abs(points1[i] - 127)) > 100:
        res1[j] = 1
    i += 1
j = 0
i = 0
while i < len(points2):
    j = int(i / 5)
    if (abs(points2[i] - 127)) > 100:
        res2[j] = 1
    i += 1
f1 = open('test2sig_answ.txt', 'w')
res1 = np.array(res1)
res2 = np.array(res2)
res = res1*res2
print(len(res))
f1.write(''.join(str(e) for e in res))
```

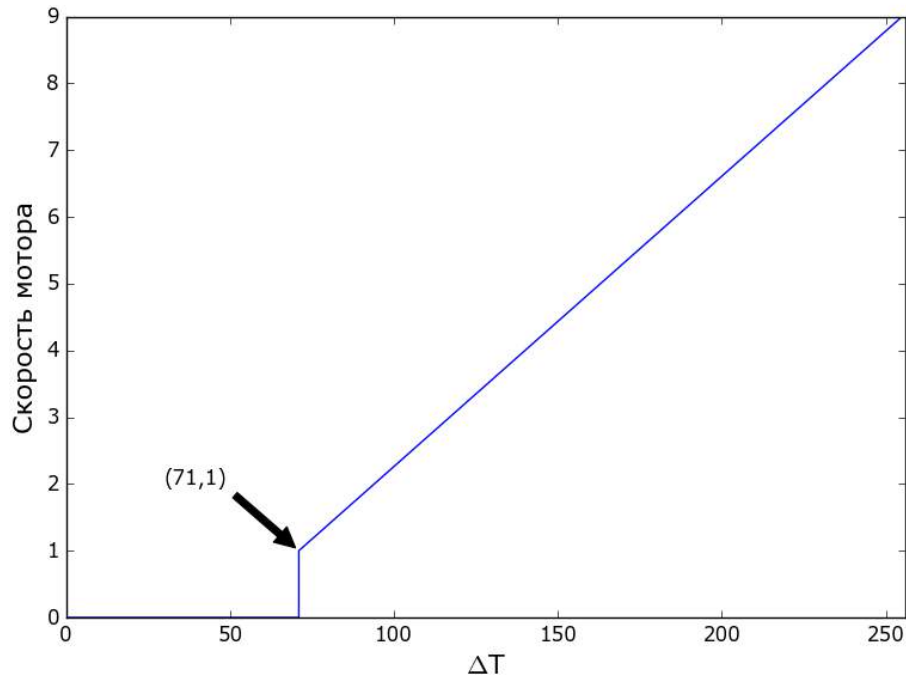
Для проверки решений используется код на языке Python, представленный в Приложении 1.

Задача 1.3 (20 баллов)

Финальная задача данного этапа позволит нам приблизиться к используемым на практике методам обработки сигналов для управления различными механизмами. Мы продолжаем работать с сигналом ЭМГ, максимальный уровень которого равен 255. Зададим окно шириной в $N=20$ точек и будем сдвигать его по длине всего сигнала с шагом в одну точку. Каждый раз будем вычислять разницу ΔT между максимальным и минимальным значением сигнала в окне. Как только это значение превысит $T_m=70$, будем включать мотор робота, задавая ему скорость вращения ротора, равную единице. Исходя из максимального значения ΔT (не всегда может быть равно 255), необходимо определить скорость вращения ротора мотора (всегда является целым числом) в каждый момент времени, считая, что она нарастает линейно по значению ΔT , и ее максимальное значение равно 9. Решением будет являться строка размером в количество точек сигнала, её m -ый элемент равен скорости вращения ротора в данный момент.

Указание: Ширина окна для первых 20 точек меняется соответственно от 1 для первой точки, до 20 для двадцатой точки. В m -ый момент времени для расчета скорости учитываются точки сигнала в диапазоне от $m-19$ до m . Для приведения скорости мотора к целому числу используйте функцию `round()`.

Пример зависимости скорости мотора от ΔT при $\Delta T_m=255$ (максимальное значение ΔT):



Решение. Ниже приведено возможное решение на языке Python 3.

```
import sys
inp = sys.stdin.readline()
points = list(map(int, inp.split(',')))
i = 0
max_level = 0
while i < len(points):
    if (i<20):
        mi = min(points[0:i+1])
        ma = max(points[0:i+1])
    else:
        mi = min(points[i-19:i+1])
        ma = max(points[i-19:i+1])
    if ma-mi > max_level:
        max_level = ma - mi
    i += 1
if max_level > 70:
    delta_t = max_level - 70
else:
    res = [0]*len(points)
    print(''.join(str(e) for e in res))
    sys.exit(0)
res = []
i = 0
while i < len(points):
    if (i<20):
        mi = min(points[0:i+1])
        ma = max(points[0:i+1])
    else:
        mi = min(points[i-19:i+1])
        ma = max(points[i-19:i+1])
    delta = ma - mi
```

```

if delta > 70:
    vel = int(round(8.0/delta_t * (delta-70))) + 1
else:
    vel = 0
res.append(vel)
i += 1
print(''.join(str(e) for e in res))

```

Для проверки решений используется код на языке Python, представленный в Приложении 1.

Блок задач №2

Для успешного выполнения задач данного блока вам необходимо познакомиться с библиотекой matplotlib:

<https://pythonworld.ru/novosti-mira-python/scientific-graphics-in-python.html>

Для наших целей будет достаточно использования интерфейса pyplot, рекомендуем ознакомиться с примером построения графиков на сайте: <https://matplotlib.org/>

В последней задаче будет использовано преобразование Фурье. Справка по функции преобразования Фурье в Python: <https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.fft.fft.html#numpy.fft.fft>

Обращаем ваше внимание на то, что анализировать необходимо лишь модуль значения полученного преобразования, для этого необходимо извлечь модуль комплексного числа, например, если (смотреть пример по ссылке выше):

```
sp = np.fft.fft(np.sin(t))
```

То по оси Y необходимо откладывать:

```

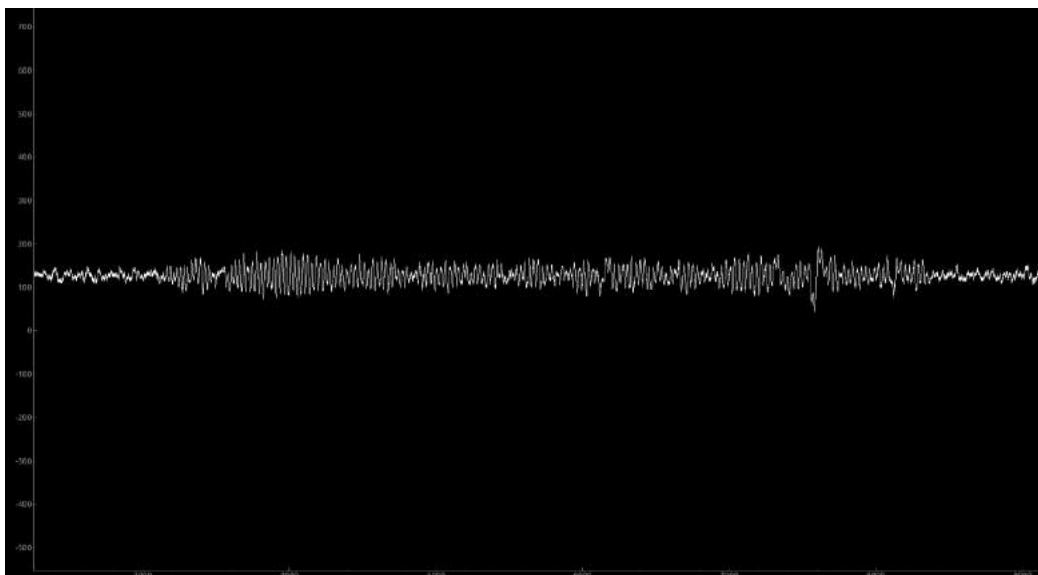
points = []
for i in sp:
    points.append(math.sqrt(math.pow(i.real,2)+math.pow(i.imag,2)))

plt.plot(freq, points)

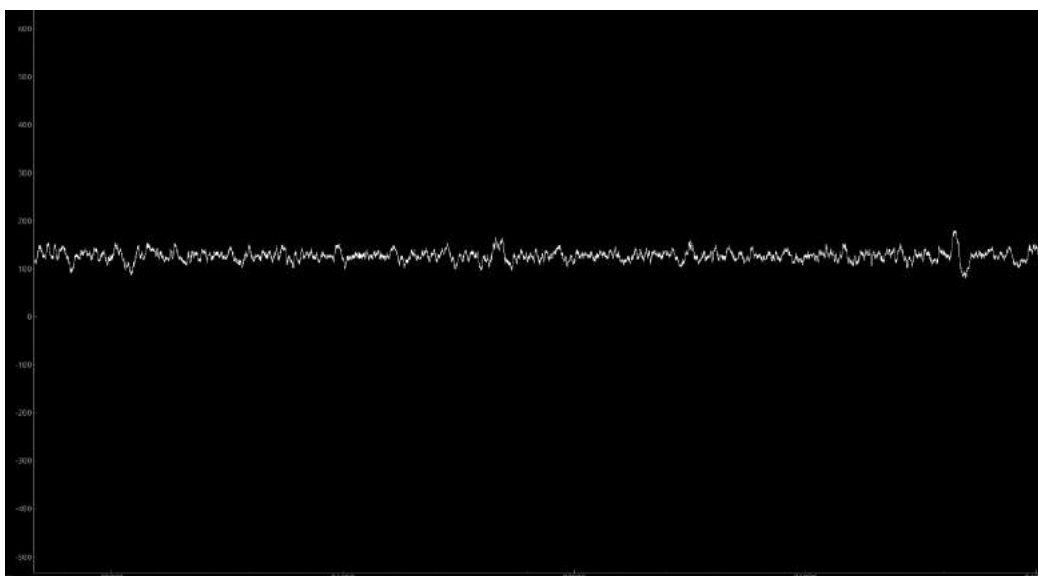
```

Задача 2.1 (10 баллов)

Для проведения различных исследований мозга иногда используют сигнал ЭЭГ. С помощью его анализа можно выявить характерные состояния, например, закрытие глаз. При закрытии глаз у большинства людей на ЭЭГ возникают колебания в диапазоне от 8 до 13 Гц, называемые альфа-ритмом. При визуализации сигнала участок с выраженным альфа-ритмом представляет из себя веретено из синусоид. Пример такого участка:



При отсутствии альфа-ритма наблюдается шум, подобный данному:



В данной задаче требуется визуализировать сигнал и определить, сколько раз человек закрывал глаза за период записи ЭЭГ. Сигнал ЭЭГ представляет из себя последовательность целых неотрицательных чисел, меньших 256. Известно, что период закрытия глаз всегда составлял более 3000 точек, а промежутки между ними не может быть меньше 4000 точек.

Файл с записью сигнала доступен по ссылке: <http://bit.ly/2hxiydW>

Для каждой команды предоставляется одна попытка вне зависимости от числа участников в ней: отправить ответ на проверку может лишь один участник от команды. При несоблюдении этого правила баллы за задачу учтены не будут.

Ответ: 5. Комментарий к ответу: При визуализации сигнала четко определяются 5 комплексов синусоидальных веретен.

Полный балл ставится за точный ответ с первой попытки от команды.

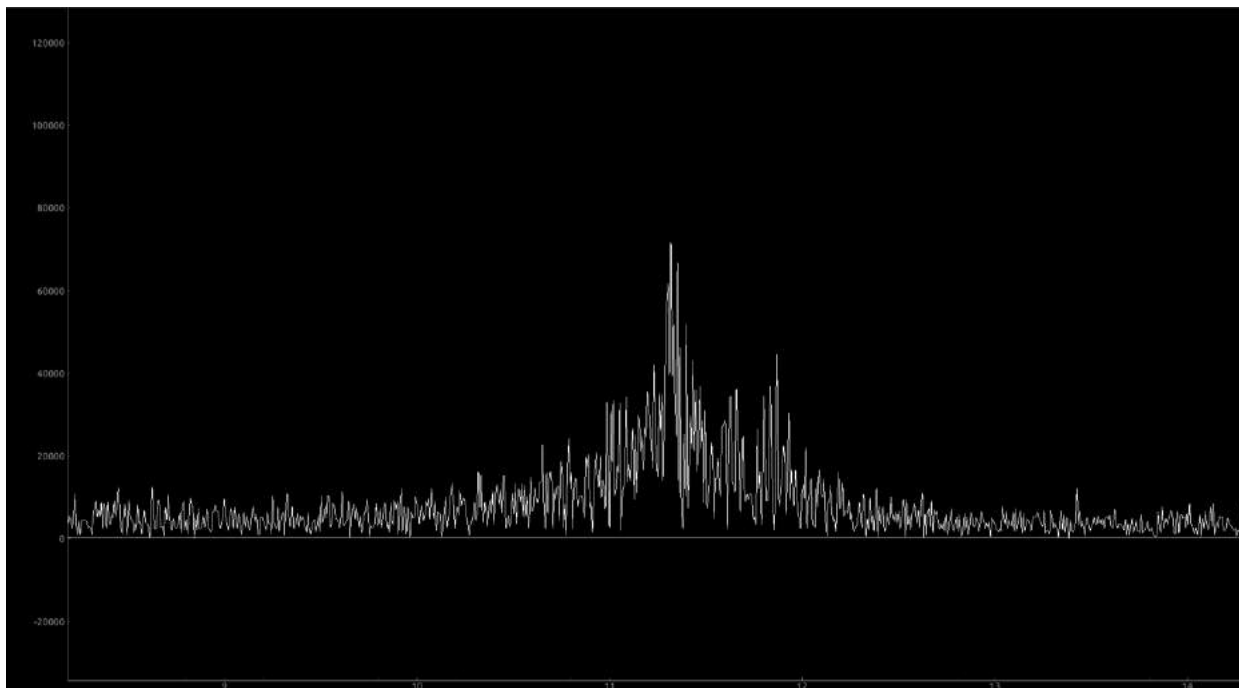
Задача 2.2 (15 баллов)

Продолжая анализировать сигнал ЭЭГ, данный в первой задаче, применим к нему преобразование Фурье, чтобы определить точную частоту пика альфа-ритма. Считая, что пик

альфа-ритма лежит в диапазоне от 8 до 13 Гц, найдем частоту в Гц, соответствующую его максимуму. Ответом будет являться число с точностью до десятых, допускающее погрешность ± 0.5 Гц.

Для каждой команды предоставляется одна попытка вне зависимости от числа участников в ней: отправить ответ на проверку может лишь один участник от команды. При несоблюдении этого правила баллы за задачу учтены не будут.

Ответ: 11.3. Комментарий к ответу: на диапазоне частот от 8 до 13 Гц наибольшую амплитуду графика наблюдается при 11.3 Гц (см. рис. ниже)



Допустимая погрешность ± 0.5 . Засчитывается правильный ответ в пределах допустимой погрешности, введенный с первой попытки от команды.

Задача 2.3 (20 баллов)

Оценим спектральную мощность сигнала альфа-ритма, данного в задаче 1. Для этого проинтегрируем Фурье-преобразование сигнала в диапазоне от 8 до 13 Гц, интегрирование будем проводить методом трапеций: будем умножать квадрат полусуммы соседних модулей комплексных чисел (значения по оси Y) на расстояние между соседними дискретными частотами (значения по оси X, в нашем случае равно 0.00680272).

Ответ допускает погрешность ± 1000 . В качестве первого теста используется сигнал из первой задачи. Количество попыток неограниченно.

Решение. Ниже приведено возможное решение на языке Python 3.

```
import sys
inp = sys.stdin.readline()
num = list(map(int, inp.split(',')))
signal = np.array(num, dtype=int)
fourier = np.fft.fft(signal)
n = signal.size
timestep = 0.003
freq = np.fft.fftfreq(n, d = timestep)
points = []
for i in fourier:
    points.append(math.sqrt(math.pow(i.real, 2) + math.pow(i.imag, 2)))
sum = 0
for i in range(0, len(freq)):
```

```
if ((freq[i]<=13)and(freq[i]>=8)):  
    sum += math.pow(points[i],2)*0.00680272  
print(str(sum))
```

Для проверки решений используется код на языке Python, представленный в Приложении 1.

Блок задач №3

Приветствуем вас в заключительной части второго этапа! Вам предстоит решить две задачи, актуальные для физиологических исследований.

В настоящее время стали очень популярны устройства, позволяющие отслеживать активность человека. Почти в каждом спортивном браслете используются данные ФПГ для определения пульса:

<https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%82%D0%BE%D0%BF%D0%BB%D0%B5%D1%82%D0%B8%D0%B7%D0%BC%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0>

Профессиональные спортсмены чаще отдают предпочтение нагрудному пульсометру, который использует подобие сигнала ЭКГ:

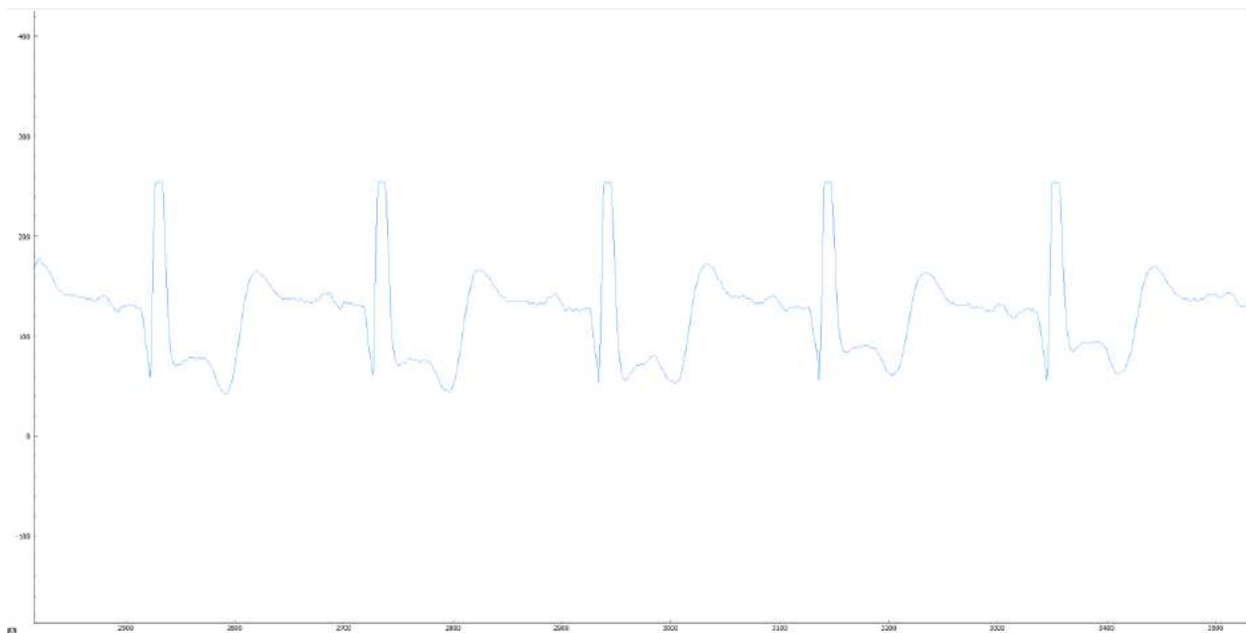
<https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BA%D0%B0%D1%80%D0%B4%D0%B8%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F>

Вам предстоит поработать с этими сигналами и познакомиться с их практическим применением.

Количество попыток ввода решений в этом блоке не ограничено.

Задача 3.1 (15 баллов)

Для повышения эффективности тренировок спортсмены могут использовать различные пульсометры. Самым точным видом пульсометров на сегодняшний день является нагрудный, он анализирует сигнал ЭКГ, вычисляя значения пульса в реальном времени. Пример такого сигнала:



В данной задаче вам предлагается найти мгновенное значение пульса, для этого на вход будет подаваться сигнал, в котором присутствуют только 2 R-зубца (самые высокие пики на изображении выше). Интервалы между R-зубцами соответствуют интервалам между ударами сердца. Сигнал на входе представляет из себя последовательность целых чисел, разделенных запятой, принадлежащих диапазону от 0 до 255. Запись сигнала ведется с частотой 330 Гц. Ответ допускает некоторую погрешность.

Входные данные для первого теста: <http://bit.ly/2B912n1>

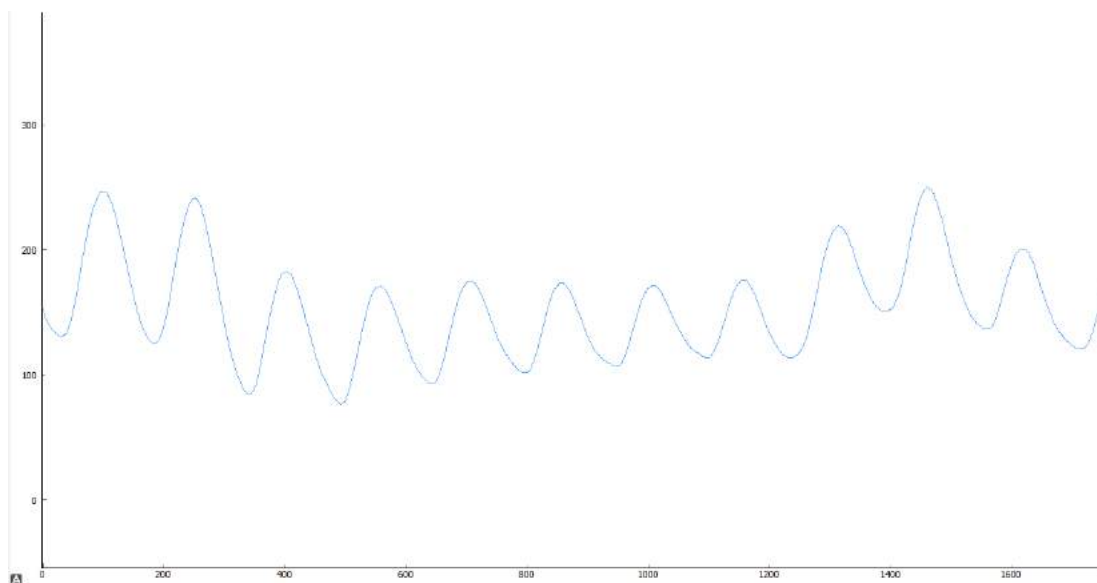
Решение. Ниже приведено возможное решение на языке Python 3.

```
import sys
inp = sys.stdin.readline()
points = list(map(int, inp.split(',')))
i = 0
length = int(len(points)/2)
max1 = np.argmax(points[:length])
max2 = np.argmax(points[length:])
max2+= length
time = (max2 - max1)/330.
num = int(60./time)
print(str(num))
```

Для проверки решений используется код на языке Python, представленный в Приложении 1.

Задача 3.2 (20 баллов)

В тех случаях, когда использование нагрудного пульсометра затруднительно, обращаются к использованию ФПГ. Этот метод менее точен, так как более подвержен окружающему воздействию. Тем не менее, в ряде случаев он дает приемлемые результаты. Пример сигнала ФПГ представлен ниже:



Пики на графике соответствуют ударам сердца. В данной задаче необходимо найти их количество за все время проведения измерений. Сигнал на выходе представляет из себя последовательность целых чисел, разделенных запятой, принадлежащих диапазону от 0 до 255.

Входные данные первого теста: <http://bit.ly/2jX5xuW>

Решение. Ниже приведено возможное решение на языке Python 3.

```
import sys
inp = sys.stdin.readline()
```

```

points = list(map(int, inp.split(', ')))
up = False
num_up = 0
num_down = 0
i = 1
num = 0
while i < len(points):
    if (points[i - 1] < points[i]):
        if (points[i - 1] < points[i]):
            if (num_up < 10):
                num_up += 1
    else:
        num_up = 0
    if num_up == 10:
        up = True
    if up and (points[i-1]>points[i]):
        if (num_down < 10):
            num_down += 1
    else:
        num_down = 0
    if num_down == 10:
        num += 1
        num_down = 0
        num_up = 0
        up = False
    i += 1
print(str(num))

```

Для проверки решений используется код на языке Python, представленный в Приложении 1.

2.2 Критерии оценивания

В первом блоке командам предстояло решить три задачи. Максимальный балл за задачи первого блока - 45 баллов максимум (Задача 1.1 - 10 баллов, Задача 1.2 - 15 баллов, Задача 1.3 - 20 баллов). У участников было неограниченное число попыток для решения этих задач. Баллы за неправильные решения не снимались. Оценка за задачу ставится пропорционально пройденным тестам. В первой задаче есть два теста, во второй - три теста, в третьей - пять тестов.

Приведем несколько примеров оценивания задач:

- Если в первой задаче решение участника пройдет только один тест из двух, то участник получит 5 баллов.
- Если в третьей задаче решение пройдет три теста из пяти, то за задачу участник получит 12 баллов.

Команда в первом блоке получала баллы за лучшие попытки участников, входящих в состав команды.

Для автоматической проверки решений использовался код на языке Python, представленный в Приложении 1.

Во втором блоке задач участникам также было предложено три задачи. Максимальный балл за задачи второго блока - 45 баллов максимум (2.1 - 10 баллов, 2.2 - 15 баллов, 2.3 - 20 баллов).

В задачах 2.1 и 2.2 для каждой команды предоставляется одна попытка вне зависимости от числа участников в ней: отправить ответ на проверку может лишь один участник от команды. При несоблюдении этого правила баллы за задачу не учитывались.

В задаче 2.1 засчитывался только точный ответ. В задаче 2.2 допустимая погрешность ± 0.5 – засчитывался правильный ответ в пределах допустимой погрешности, введенный с первой попытки от команды.

В задаче 2.3 баллы ставятся пропорционально пройденным тестам (как в первом блоке задач).

Для автоматической проверки решений использовался код на языке Python, представленный в Приложении 1.

В **третьем** блоке участникам было предложено две задачи (3.1 – 15 баллов, 3.2 – 20 баллов). Количество попыток ввода решений в этом блоке не ограничено. Учитывался результат лучшей попытки.

Для автоматической проверки решений использовался код на языке Python, представленный в Приложении 1:

<http://nti-contest.ru/wp-content/uploads/neuro-2018.pdf>

.