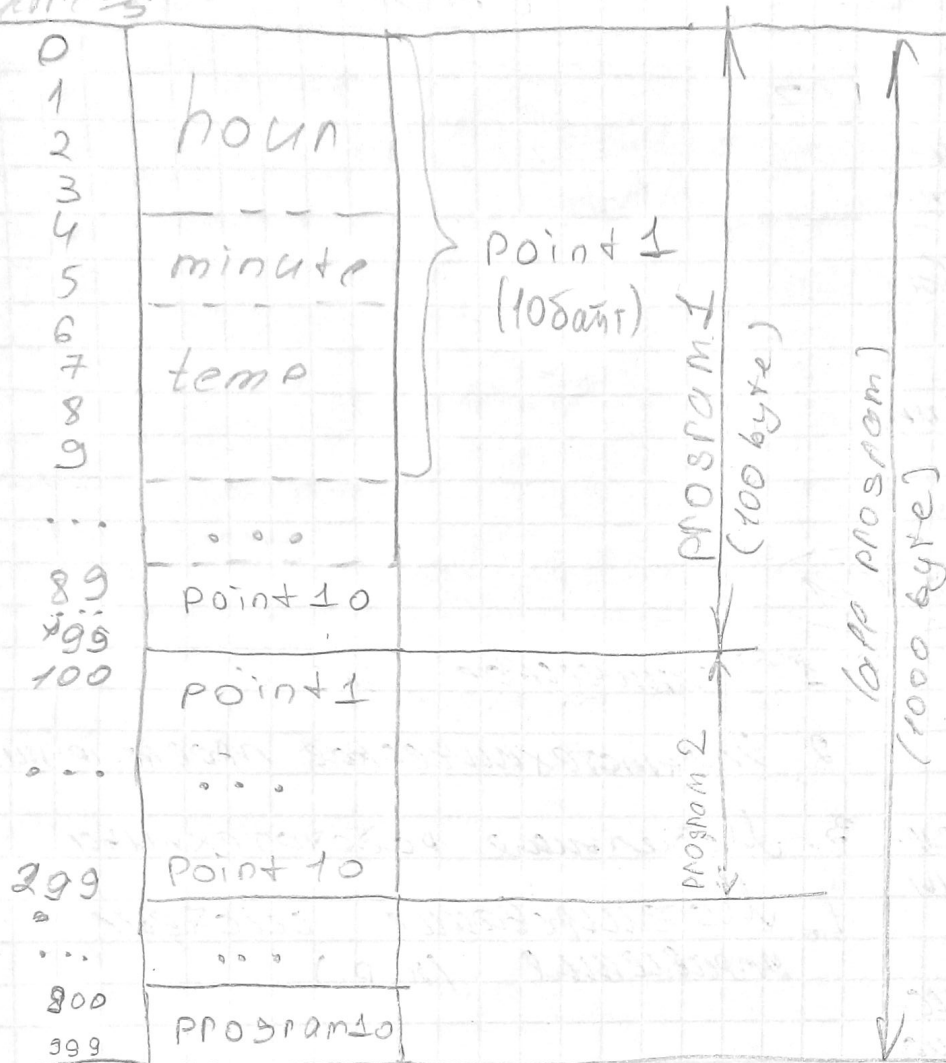


# EEPROM

address



Адрес начала  
структуры point i

$$A = (i - 1) \times 10$$

Адрес начала  
программы  
программ i i = 1

$$A = (i - 1) \times 100$$

# Участок READ-POINTS

переменные:  
byte i = 0;

ПАРСИНГ строки  
Нахождение N, P  
int | int

Входные  
данные:

#0 0;

D = N, P

N - номер пр-ммы

P - номер страницы

~~Чтение~~ Вычисление  
адреса начала  
в EEPROM, где находится  
начало программы.

$$a = (N - 1) \times 100$$

да  $P = 0$  нет ( $1 \neq P$ )

считываем  
первые  
5 точек

i = 0

Читаем  
структуру  
с адресом a

последовательно  
отправляем  
данные в эмулятор

переходим  
к следующей точке  
(вычисляем новый  
адрес  $a += 10$ .)

i++

i < 5

нет

считываем 6-10  
точки

вычисляем новый  
адрес

$$a += 50$$

(переходим к 6 точке)

по аналогии

конец



Структура для записи  
точки программы

```
struct point {  
    char[4] hour; (4 байт)  
    char[2] minute;  
    char[4] temp; } размер 10 байт
```

Массив объектов таких структур  
100 штук (размер 1000 байт)



## READ-POINT

на входе

$$\text{last} = D = \underbrace{N, P}_{\text{символов}}$$

символов

Переменные:  $\text{int } P$

$\text{int } A, P, \&\text{spcc}$

Парсим строку и last  
Находим  $N, P$

Формируем адрес нужной точки  
(начало нужной структуры)

$$A = \underbrace{(N-1) \times 100}_{\text{начало программы}} + \underbrace{(P-1) \times 10}_{\text{начало точки}}$$

Ищем точку из памяти

Записываем элементы структуры  
в дисплей

конец



## WRITE\_POINTS

На входе:  $cast = D = N, P, H, M, T$   
15 символов

Разбираем строку  
N, P - переводим в  
число

Формируем  
адрес точки

$$a = (N-1) \times 100 + (P-1) \times 10$$

Записываем последовательно  
H, M, T в элементы структуры  
point

Отправляем структуру point  
по адресу a

конец