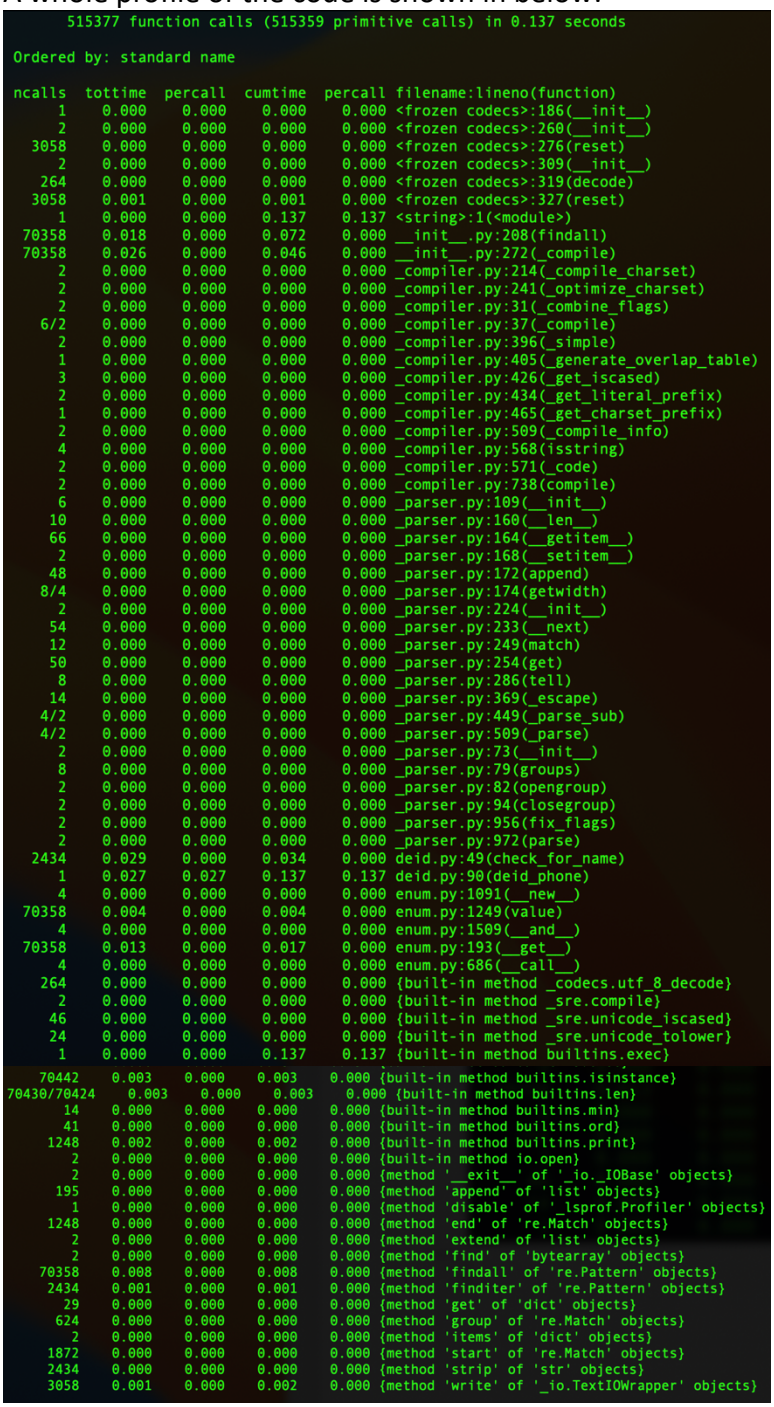
The screenshots of code output (part).

```
1 7 538 556 Started Dobutamine
1 15 470 487 Requesting Miller
1 28 636 646 Dr Vasquez
1 32 1 17 Nursing Progress
1 32 528 541 Heart Failure
1 34 1 17 Nursing Progress
1 34 358 369 Await Psych
1 34 1689 1698 Full Code
1 37 114 120 Per Dr
1 38 225 241 Given Mesalamine
1 38 248 258 Aloe Vesta
1 39 1115 1128 Titrate Lasix
1 50 91 108 Cardic Transplant
1 53 393 403 Dr Vasquez
1 66 204 214 Dr Vasquez
1 72 255 265 Holy Cross
2 13 42 53 Harbor Hosp
3 1 0 9 Resp Care
3 6 626 633 On Levo
3 9 5 21 Nursing Progress
3 9 1634 1643 Full Code
3 9 1655 1667 Husband Rich
3 9 1730 1739 San Diego
3 12 1 17 Respiratory Care
3 12 98 111 Bernard Foley
3 17 235 242 If Dopa
3 20 1 10 Resp Care
3 23 1 10 Resp Care
3 25 1 17 Respiratory Care
3 29 1 17 Respiratory Care
3 37 1 17 Respiratory Care
3 41 1 17 Respiratory Care
5 1 0 16 Respiratory Care
5 3 0 9 Resp Care
5 7 0 9 Resp Care
5 14 1 17 Respiratory Care
5 28 1 17 Respiratory Care
5 29 1 17 Respiratory Care
7 2 714 729 Currently Nitro
8 1 537 550 Carol Buckley
8 4 130 146 General Hospital
8 4 147 161 Medical Center
8 4 962 977 Marcela Carlson
9 1 259 265 Was On
10 1 1 17 Respiratory Care
11 1 123 137 Adventist Hosp
11 1 1029 1042 Restarted Neo
11 3 38 55 Kessler Adventist
13 9 88 103 Started Zestril
13 11 57 73 Changed Atenolol
14 5 1 10 Resp Care
14 10 8 17 Care Note
15 1 480 493 Cor Pulmonale
15 2 1628 1639 Van Leeuwen
15 4 7 16 Care Note
15 8 1 10 Resp Care
15 11 618 632 Received Lasix
15 11 661 671 Ionized Ca
15 11 806 813 Big Boy
15 13 2 11 Resp Care
15 16 2 11 Resp Care
15 17 817 830 Mrs Nicholson
15 19 2 11 Resp Care
15 20 528 537 Dr Marder
15 20 744 761 Calcium Gluconate
15 22 2 11 Resp Care
15 24 8 17 Care Note
15 26 2 11 Resp Care
15 28 8 17 Care Note
15 31 2 18 Respiratory Care
15 31 41 55 Assist Control
15 32 2 11 Resp Care
15 34 2 18 Respiratory Care
15 47 129 140 Weaned Peep
15 47 175 184 Resp Rate
15 49 647 660 Sweeney Olsen
15 50 567 578 Start Nepro
15 50 795 814 Transplant Resident
15 53 2 11 Resp Care
15 61 2 11 Resp Care
15 64 2 11 Resp Care
15 65 2 11 Resp Care
15 66 871 884 Dr Swackhamer
15 72 2 11 Resp Care
15 73 366 376 Holy Cross
15 76 2 11 Resp Care
15 80 2 11 Resp Care
15 90 426 436 Time Frame
15 108 3 19 Respiratory Care
15 110 66 76 Passy Muir
15 116 3 12 Resp Care
15 119 3 12 Resp Care
15 124 3 12 Resp Care
15 128 3 12 Resp Care
15 131 57 69 Sacred Heart
15 132 3 12 Resp Care
15 137 192 204 Sacred Heart
15 137 305 317 Sacred Heart
15 139 3 16 Pmicu Nursing
15 139 17 30 Progress Note
15 140 28 40 Sacred Heart
16 1 353 371 Carpenter Assisted
16 5 1 10 Resp Care
16 10 2 11 Resp Care
16 13 2 11 Resp Care
16 18 2 11 Resp Care
16 20 8 17 Care Note
16 23 2 11 Resp Care
16 24 694 706 Remains Afeb
16 25 346 357 Vital Signs
16 25 1308 1318 Time Frame
16 26 8 17 Care Note
16 31 2 18 Respiratory Care
16 33 2 11 Resp Care
16 34 190 201 Vital Signs
16 34 264 277 Breath Sounds
16 34 783 793 Time Frame
16 35 2 18 Respiratory Care
```

A whole profile of the code is shown in below:



```
      515377 function calls (515359 primitive calls) in 0.137 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     1    0.000    0.000    0.000    0.000 <frozen codecs>:186(__init__)
     2    0.000    0.000    0.000    0.000 <frozen codecs>:260(__init__)
  3058    0.000    0.000    0.000    0.000 <frozen codecs>:276(reset)
     2    0.000    0.000    0.000    0.000 <frozen codecs>:309(__init__)
   264    0.000    0.000    0.000    0.000 <frozen codecs>:319(decode)
  3058    0.001    0.000    0.001    0.000 <frozen codecs>:327(reset)
     1    0.000    0.000    0.137    0.137 <string>:1(<module>)
 70358    0.018    0.000    0.072    0.000 __init__.py:208(findall)
 70358    0.026    0.000    0.046    0.000 __init__.py:272(_compile)
     2    0.000    0.000    0.000    0.000 _compiler.py:214(_compile_charset)
     2    0.000    0.000    0.000    0.000 _compiler.py:241(_optimize_charset)
     2    0.000    0.000    0.000    0.000 _compiler.py:31(_combine_flags)
   6/2    0.000    0.000    0.000    0.000 _compiler.py:37(_compile)
     2    0.000    0.000    0.000    0.000 _compiler.py:396(_simple)
     1    0.000    0.000    0.000    0.000 _compiler.py:405(_generate_overlap_table)
     3    0.000    0.000    0.000    0.000 _compiler.py:426(_get_iscased)
     2    0.000    0.000    0.000    0.000 _compiler.py:434(_get_literal_prefix)
     1    0.000    0.000    0.000    0.000 _compiler.py:465(_get_charset_prefix)
     2    0.000    0.000    0.000    0.000 _compiler.py:509(_compile_info)
     4    0.000    0.000    0.000    0.000 _compiler.py:568(isstring)
     2    0.000    0.000    0.000    0.000 _compiler.py:571(_code)
     2    0.000    0.000    0.000    0.000 _compiler.py:738(compile)
     6    0.000    0.000    0.000    0.000 _parser.py:109(__init__)
    10    0.000    0.000    0.000    0.000 _parser.py:160(__len__)
    66    0.000    0.000    0.000    0.000 _parser.py:164(__getitem__)
     2    0.000    0.000    0.000    0.000 _parser.py:168(__setitem__)
    48    0.000    0.000    0.000    0.000 _parser.py:172(append)
   8/4    0.000    0.000    0.000    0.000 _parser.py:174(getwidth)
     2    0.000    0.000    0.000    0.000 _parser.py:224(__init__)
    54    0.000    0.000    0.000    0.000 _parser.py:233(__next)
    12    0.000    0.000    0.000    0.000 _parser.py:249(match)
    50    0.000    0.000    0.000    0.000 _parser.py:254(get)
     8    0.000    0.000    0.000    0.000 _parser.py:286(tell)
    14    0.000    0.000    0.000    0.000 _parser.py:369(_escape)
   4/2    0.000    0.000    0.000    0.000 _parser.py:449(_parse_sub)
   4/2    0.000    0.000    0.000    0.000 _parser.py:509(_parse)
     2    0.000    0.000    0.000    0.000 _parser.py:73(__init__)
     8    0.000    0.000    0.000    0.000 _parser.py:79(groups)
     2    0.000    0.000    0.000    0.000 _parser.py:82(opengroup)
     2    0.000    0.000    0.000    0.000 _parser.py:94(closegroup)
     2    0.000    0.000    0.000    0.000 _parser.py:956(fix_flags)
     2    0.000    0.000    0.000    0.000 _parser.py:972(parse)
  2434    0.029    0.000    0.034    0.000 deid.py:49(check_for_name)
     1    0.027    0.027    0.137    0.137 deid.py:90(deid_phone)
     4    0.000    0.000    0.000    0.000 enum.py:1091(__new__)
 70358    0.004    0.000    0.004    0.000 enum.py:1249(value)
     4    0.000    0.000    0.000    0.000 enum.py:1509(__and__)
 70358    0.013    0.000    0.017    0.000 enum.py:193(__get__)
     4    0.000    0.000    0.000    0.000 enum.py:686(__call__)
   264    0.000    0.000    0.000    0.000 {built-in method _codecs.utf_8_decode}
     2    0.000    0.000    0.000    0.000 {built-in method _sre.compile}
    46    0.000    0.000    0.000    0.000 {built-in method _sre.unicode_iscased}
    24    0.000    0.000    0.000    0.000 {built-in method _sre.unicode_tolower}
     1    0.000    0.000    0.137    0.137 {built-in method builtins.exec}
 70442    0.003    0.000    0.003    0.000 {built-in method builtins.isinstance}
70430/70424    0.003    0.000    0.003    0.000 {built-in method builtins.len}
    14    0.000    0.000    0.000    0.000 {built-in method builtins.min}
    41    0.000    0.000    0.000    0.000 {built-in method builtins.ord}
  1248    0.002    0.000    0.002    0.000 {built-in method builtins.print}
     2    0.000    0.000    0.000    0.000 {built-in method io.open}
     2    0.000    0.000    0.000    0.000 {method '__exit__' of '_io._IOBase' objects}
   195    0.000    0.000    0.000    0.000 {method 'append' of 'list' objects}
     1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
  1248    0.000    0.000    0.000    0.000 {method 'end' of 're.Match' objects}
     2    0.000    0.000    0.000    0.000 {method 'extend' of 'list' objects}
     2    0.000    0.000    0.000    0.000 {method 'find' of 'bytearray' objects}
 70358    0.008    0.000    0.008    0.000 {method 'findall' of 're.Pattern' objects}
  2434    0.001    0.000    0.001    0.000 {method 'finditer' of 're.Pattern' objects}
    29    0.000    0.000    0.000    0.000 {method 'get' of 'dict' objects}
   624    0.000    0.000    0.000    0.000 {method 'group' of 're.Match' objects}
     2    0.000    0.000    0.000    0.000 {method 'items' of 'dict' objects}
  1872    0.000    0.000    0.000    0.000 {method 'start' of 're.Match' objects}
  2434    0.000    0.000    0.000    0.000 {method 'strip' of 'str' objects}
  3058    0.001    0.000    0.002    0.000 {method 'write' of '_io.TextIOWrapper' objects}
```

Different columns represent different properties.

ncalls: The number of times each function was called.

tottime: The total time spent inside the function.

percall: The time spent per call (equal to tottime divided by ncalls).

cumtime: The cumulative time spent in the function and its sub-functions (including function calls).

percall: The time spent per call (equal to cumtime divided by ncalls).