Pontifícia Universidade Católica do Paraná

Escola Politécnica – Bacharelado em Ciência da Computação – 7o Período

Prof. Edson Emílio Scalabrin

Nome: Gustavo Hammerschmidt.

**Trabalho:** 01

**Data de entrega:** quinta-feira seguinte a 1a avaliação somativa.

**Grupo:** até 5 integrantes.

**Peso na nota do RA01:** 40%

**Descrição do problema:** Dado 50 números inteiros de 1 a 50 e as seguintes combinações.

= 2.118.760

= 230.300

= 19.600

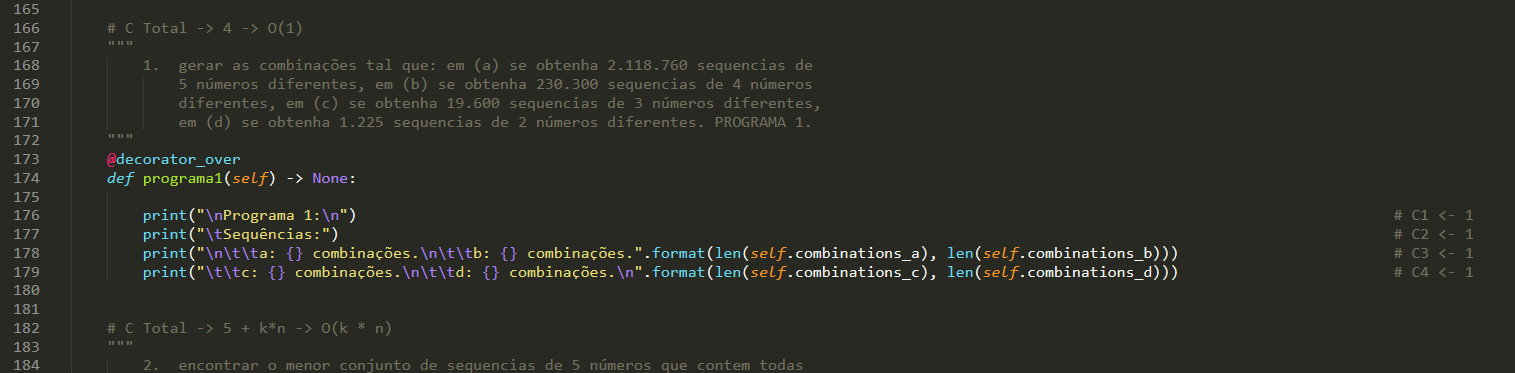
= 1.225

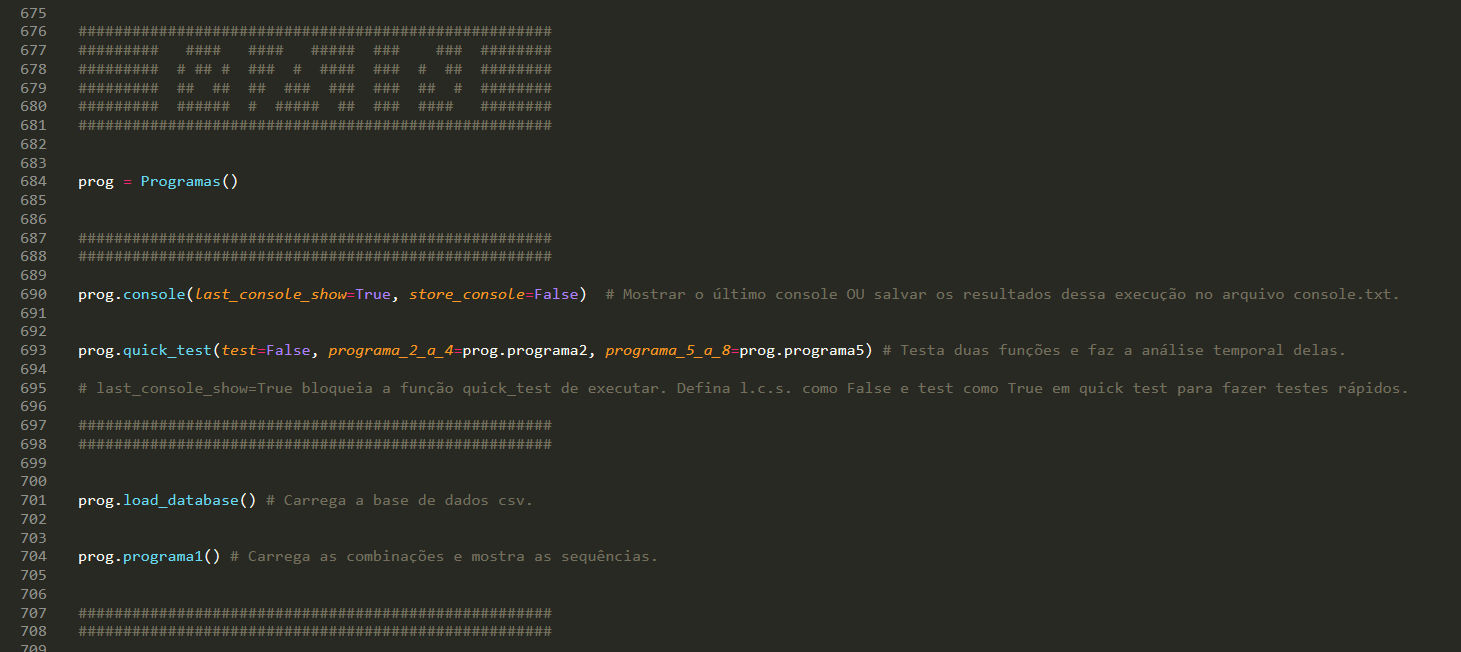
Dica: acompanhe o código em python (trabalho1.py) e output gerado e salvo (PRIME\_console.txt) junto das questões a seguir.

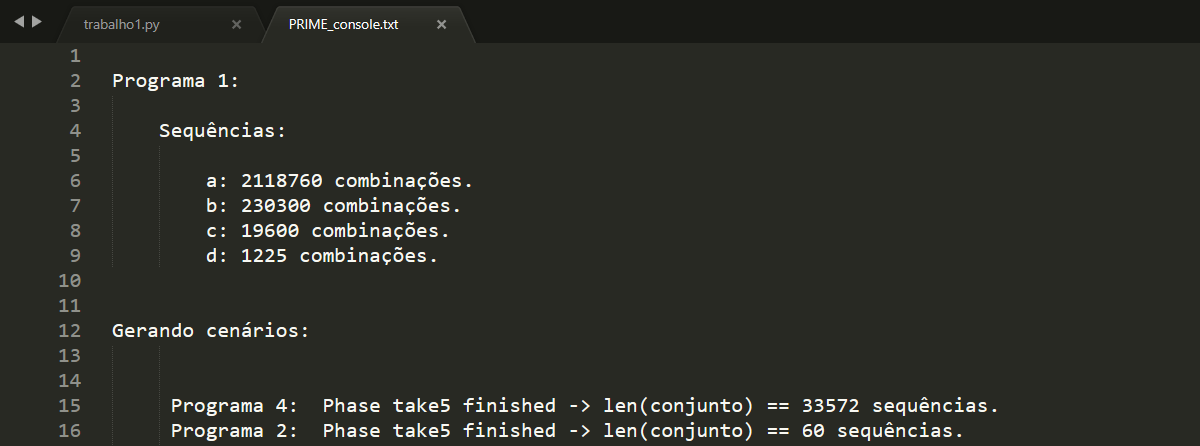
Pede-se para:

1. gerar as combinações tal que: em (a) se obtenha 2.118.760 sequencias de 5 números diferentes, em (b) se obtenha 230.300 sequencias de 4 números diferentes, em (c) se obtenha 19.600 sequencias de 3 números diferentes, em (d) se obtenha 1.225 sequencias de 2 números diferentes. PROGRAMA 1.

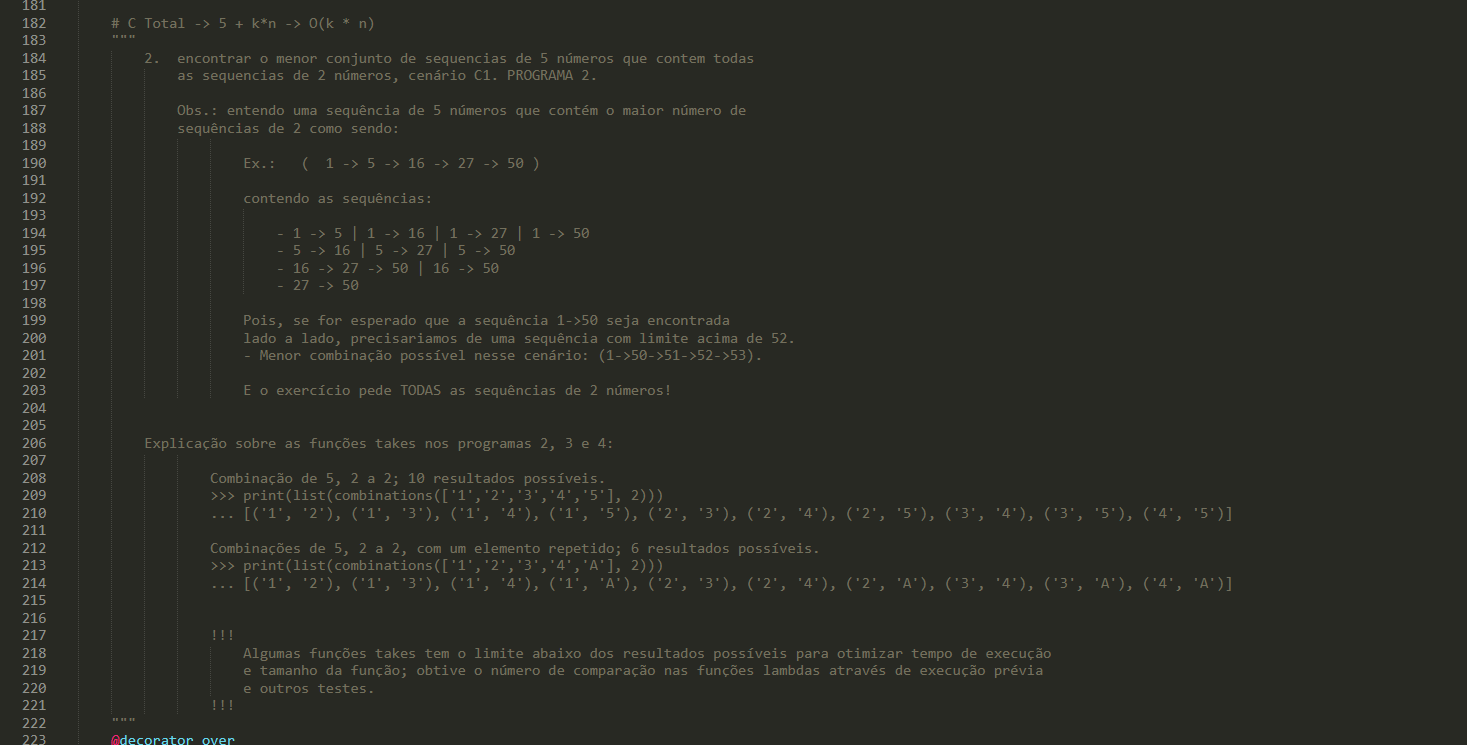


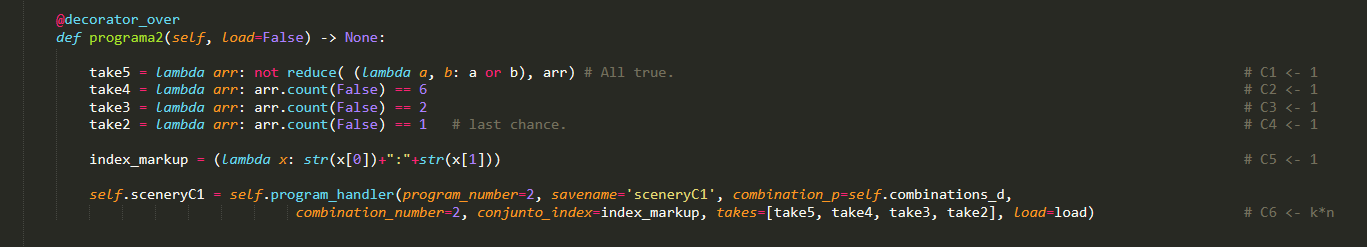


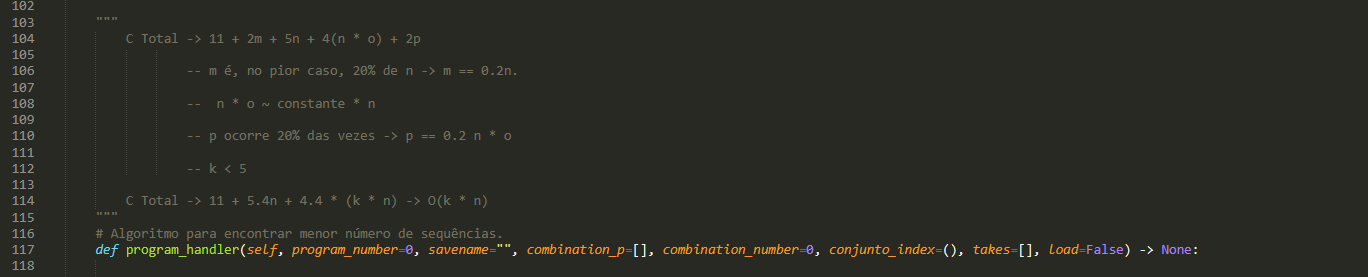


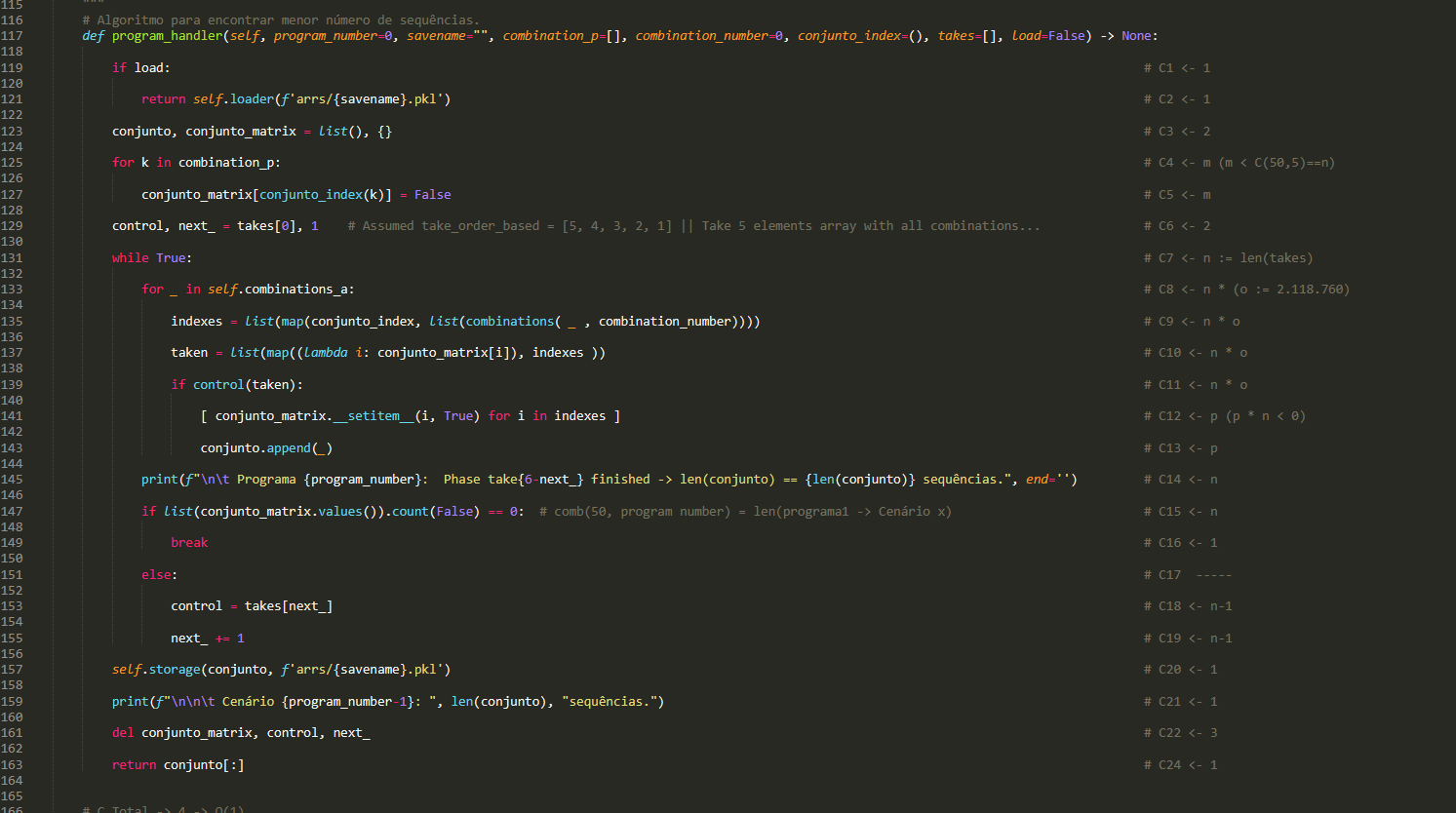


1. encontrar o menor conjunto de sequencias de 5 números que contém todas as sequencias de 2 números, cenário C1. PROGRAMA 2.

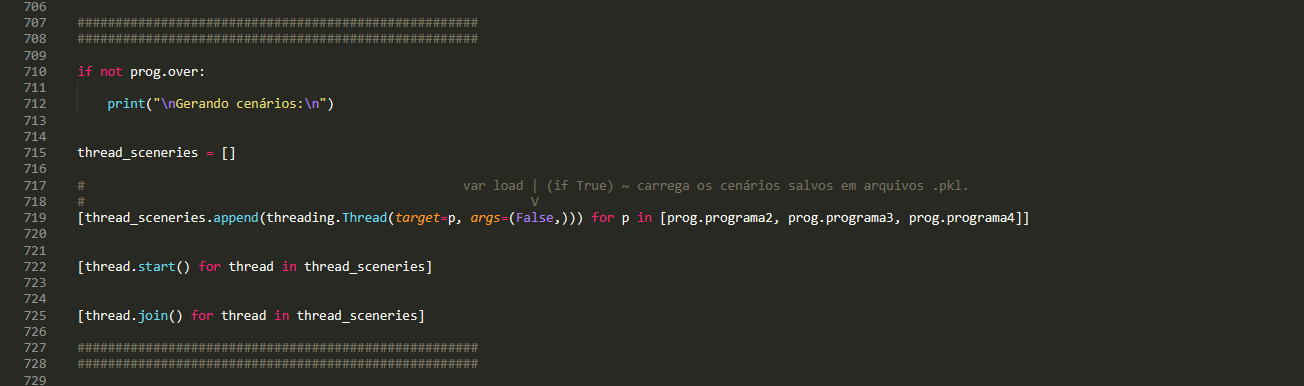


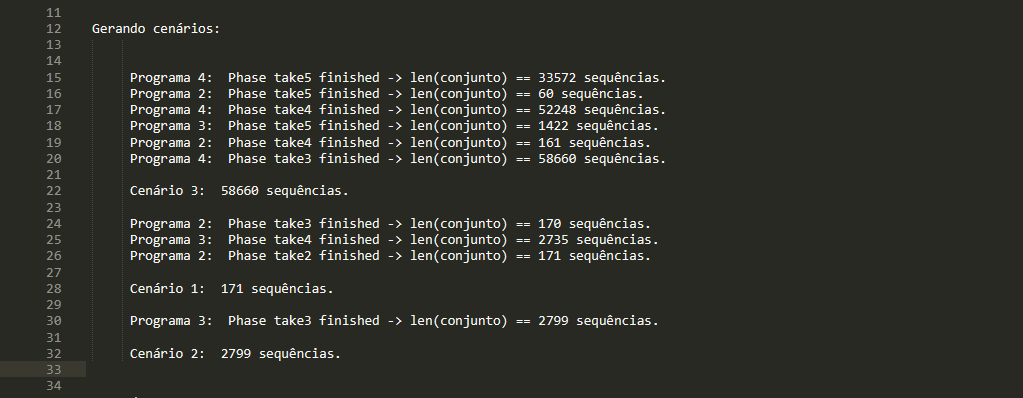




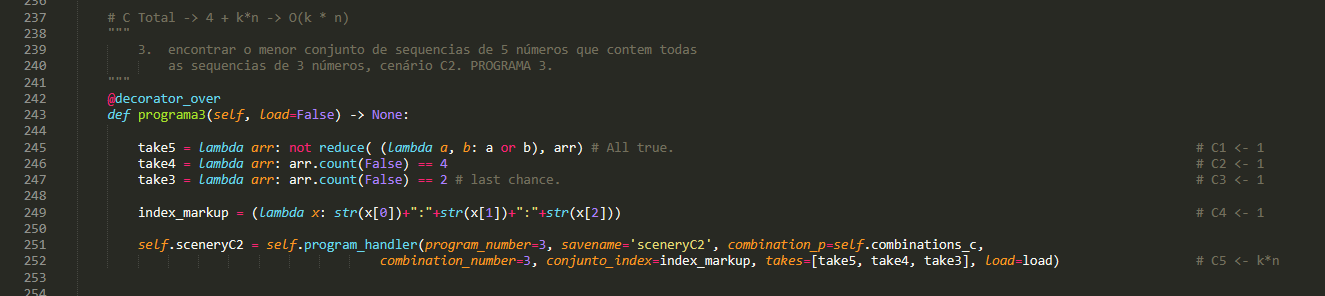


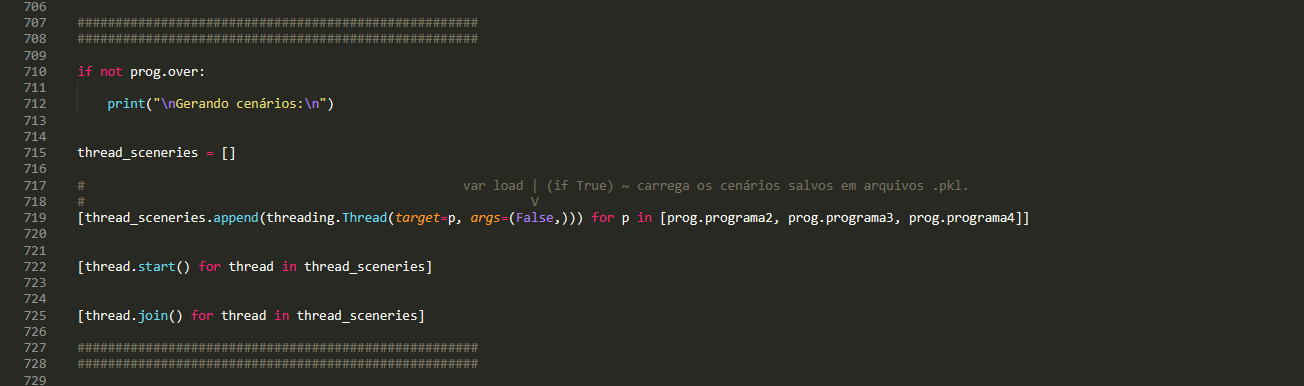
Programas 3 e 4 chamam essa função também.

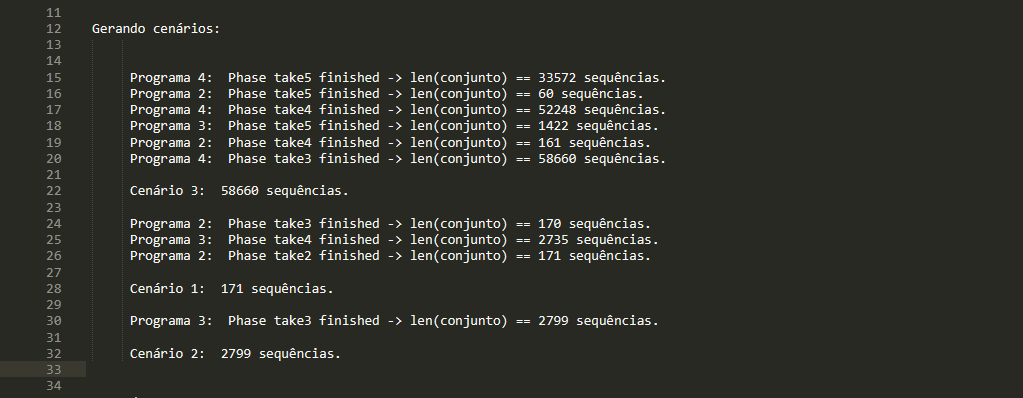




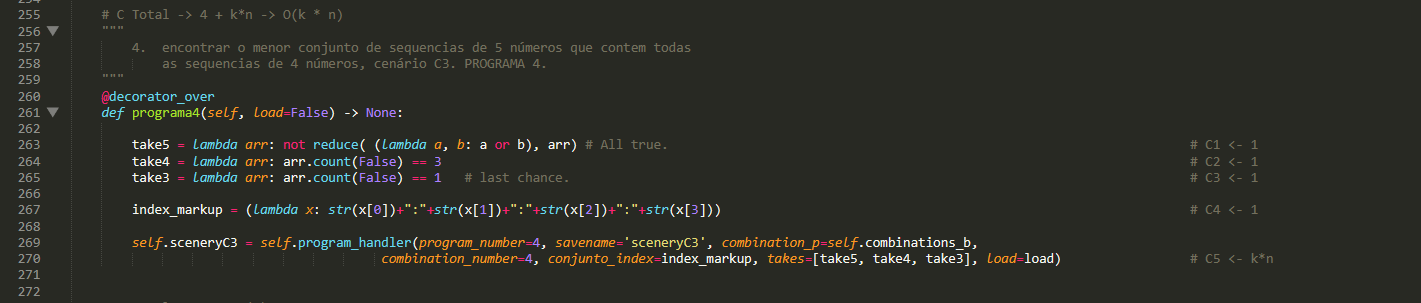
1. encontrar o menor conjunto de sequencias de 5 números que contem todas as sequencias de 3 números, cenário C2. PROGRAMA 3.

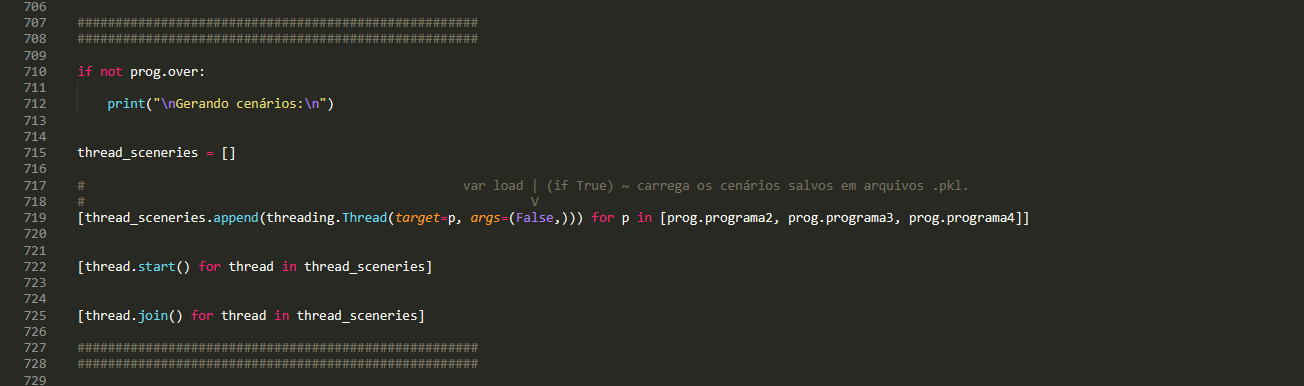


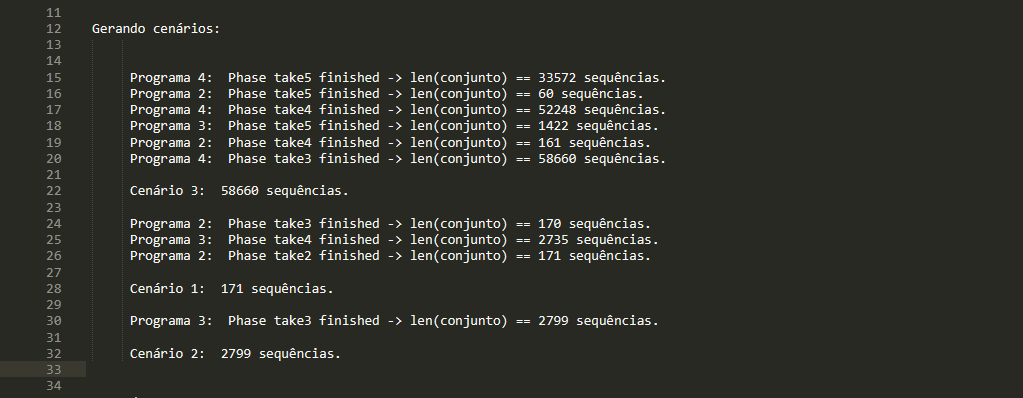




1. encontrar o menor conjunto de sequencias de 5 números que contem todas as sequencias de 4 números, cenário C3. PROGRAMA 4.



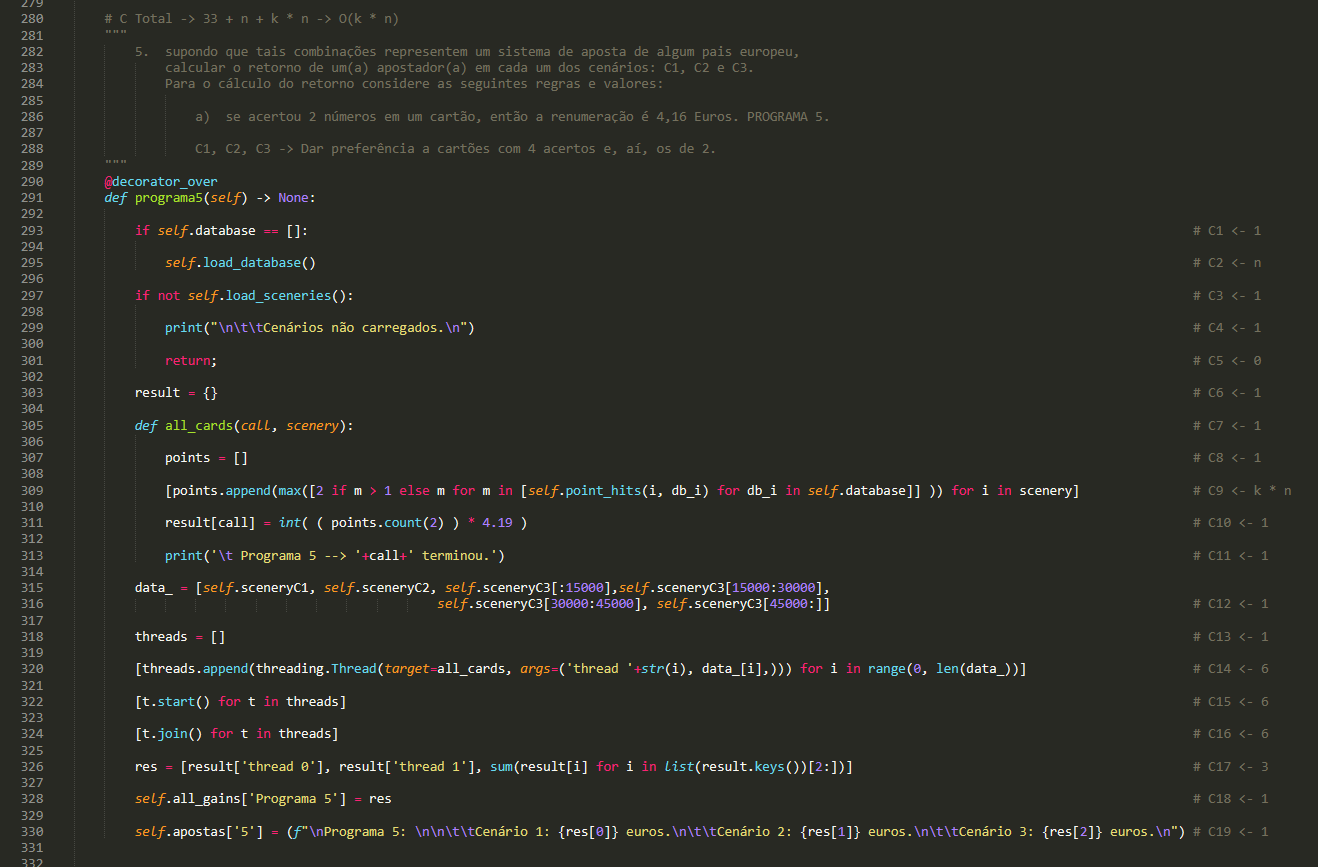




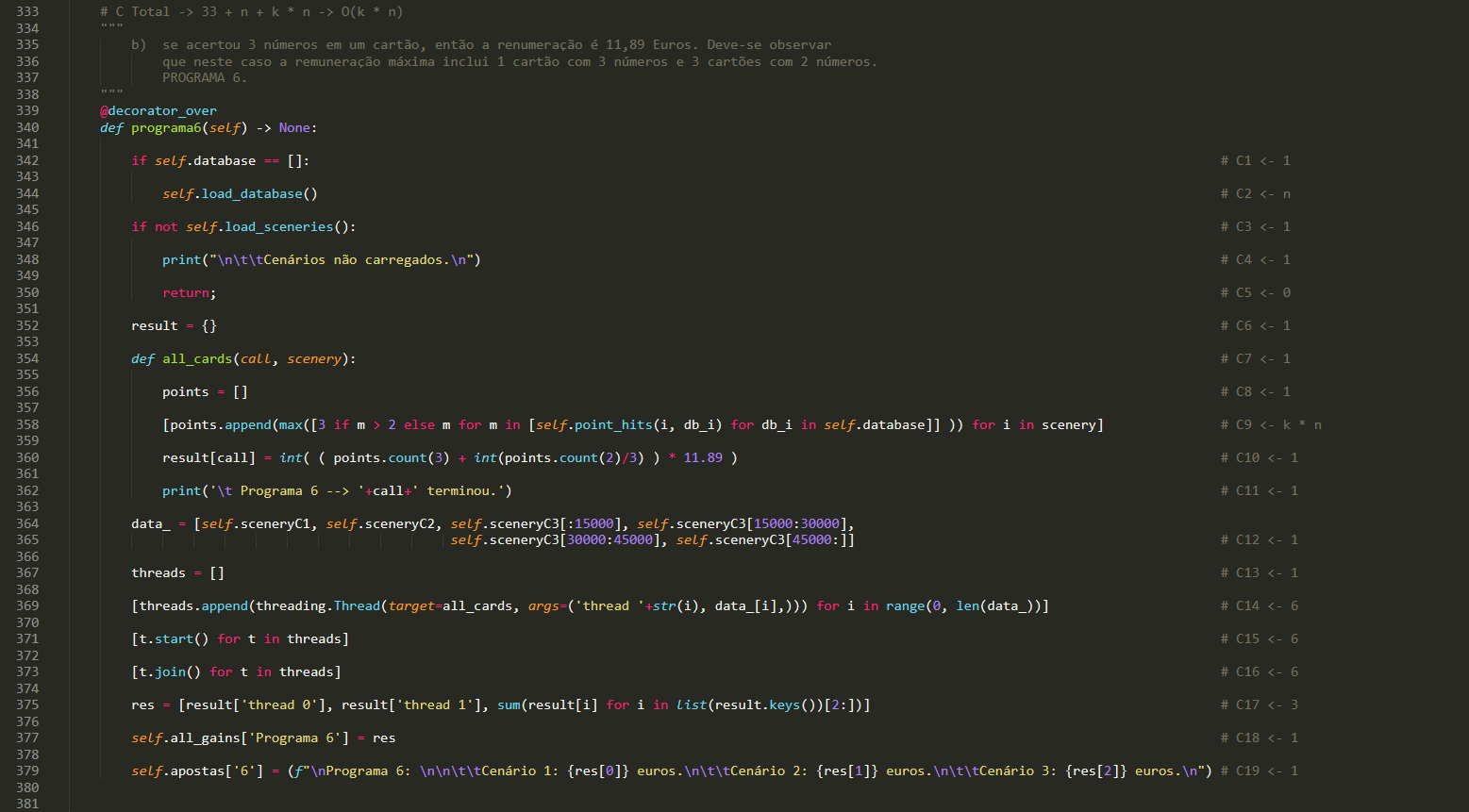
1. supondo que tais combinações representem um sistema de aposta de algum pais europeu, calcular o retorno de um(a) apostador(a) em cada um dos cenários: C1, C2 e C3. Para o cálculo do retorno considere as seguintes regras e valores:

Resultados do item 5 depois dos items: a, b, c e d.

1. se acertou 2 números em um cartão, então a renumeração é 4,16 Euros. PROGRAMA 5.



1. se acertou 3 números em um cartão, então a renumeração é 11,89 Euros. Deve-se observar que neste caso a remuneração máxima inclui 1 cartão com 3 números e 3 cartões com 2 números. PROGRAMA 6.

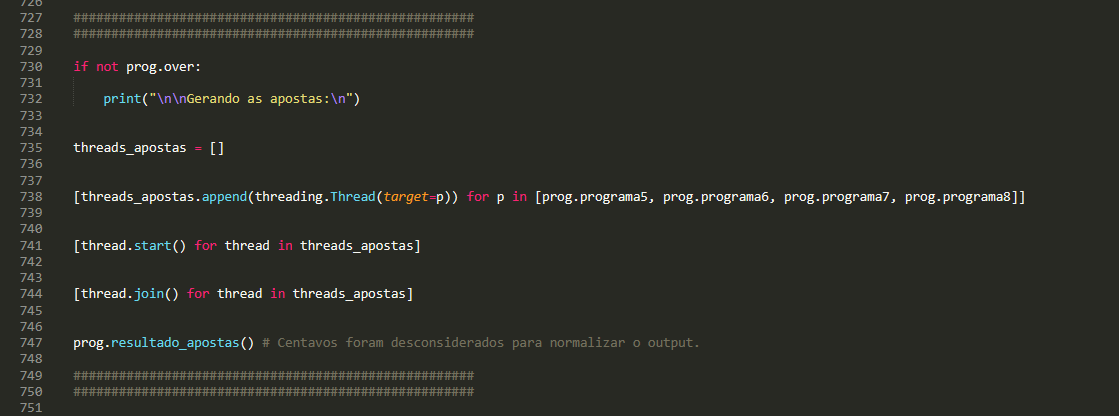


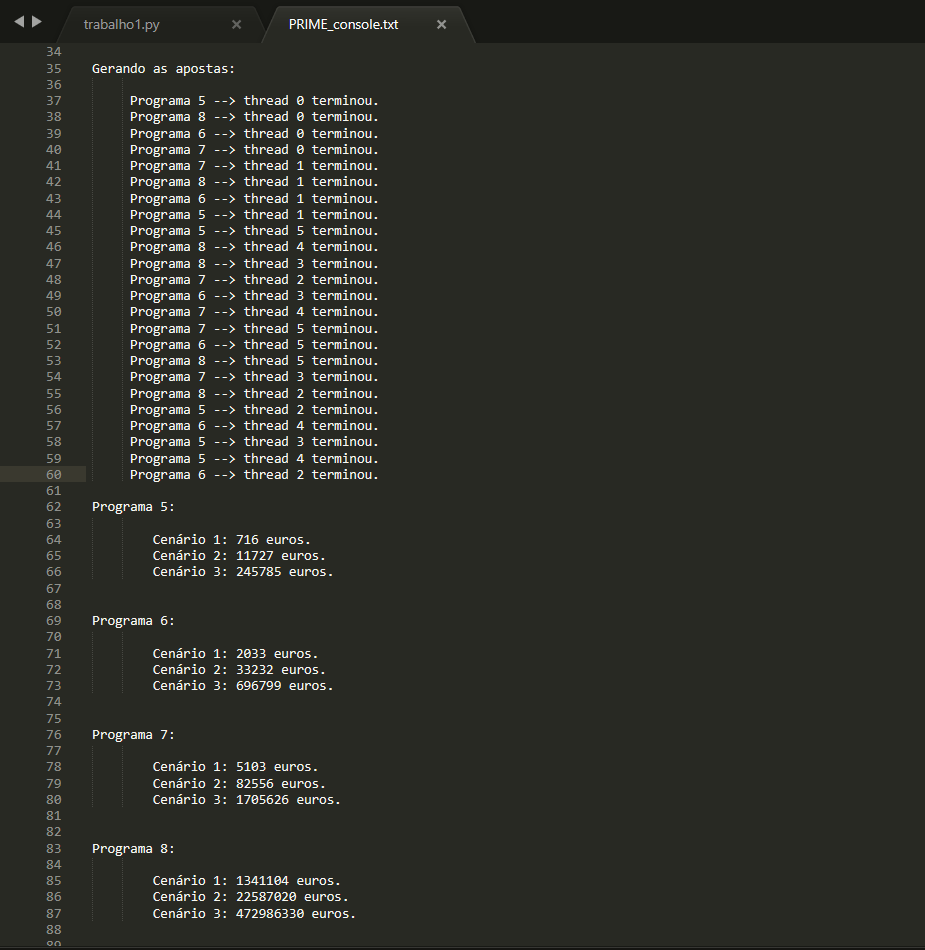
1. se acertou 4 números em um cartão, então a renumeração é 82,31 Euros. Deve-se observar que neste caso a remuneração máxima inclui 1 cartão com 4 números, 4 cartões com 3 números e 6 cartões com 2 números. PROGRAMA 7.



1. se acertou 5 números em um cartão, então a renumeração é 70.584,44 Euros. Deve-se observar que neste caso a remuneração máxima inclui 1 cartão com 5 números, 5 cartões com 4 números, 10 cartões com 3 números, 10 cartões com 2 números. PROGRAMA 8.



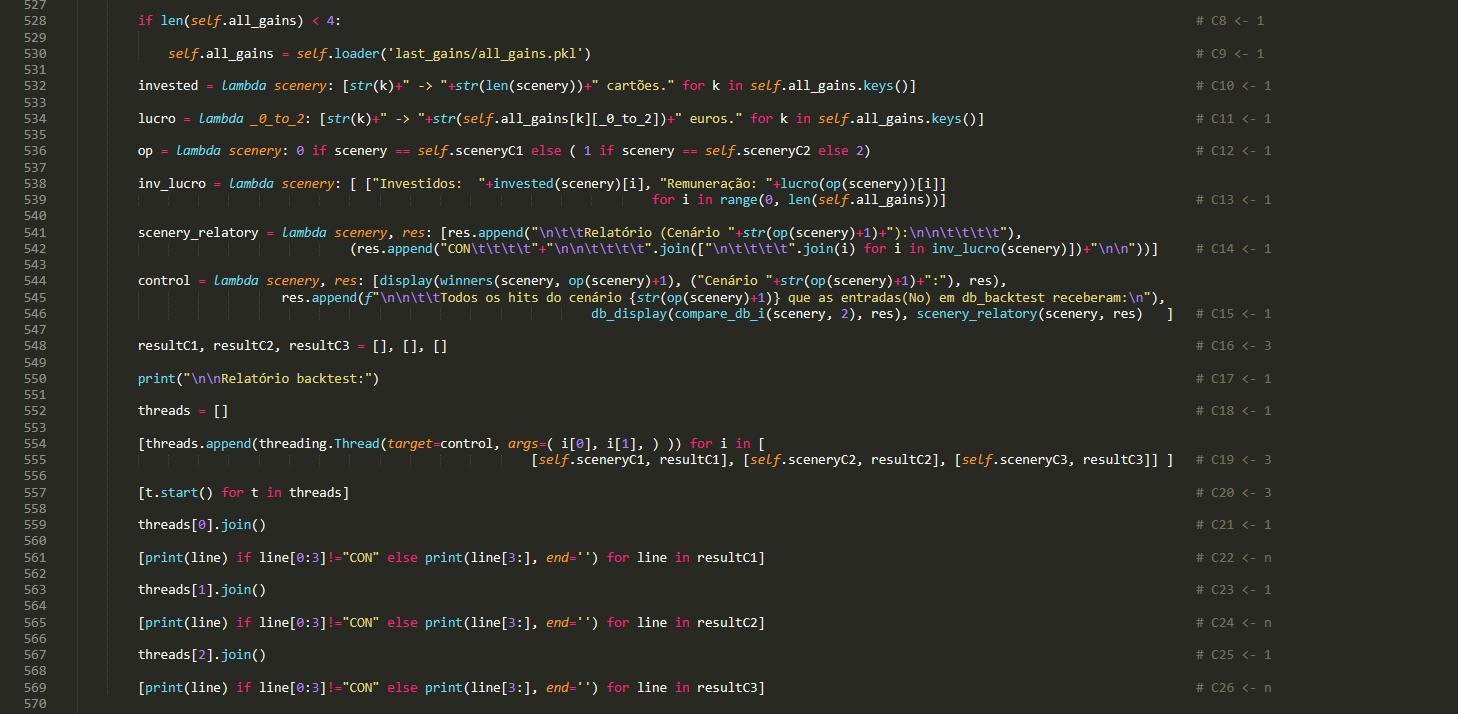


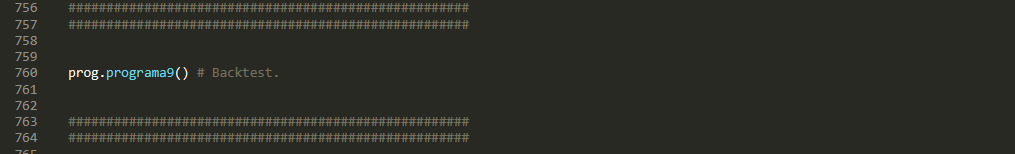


1. Tomando como exemplo o histórico de prêmios dos últimos anos disponíveis no “euro\_concursos.csv”, escreva um programa de *backtest* e gere um relatório que mostre os valores investidos e os retornos obtidos. PROGRAMA 9.

Primeiro: função do programa 9.





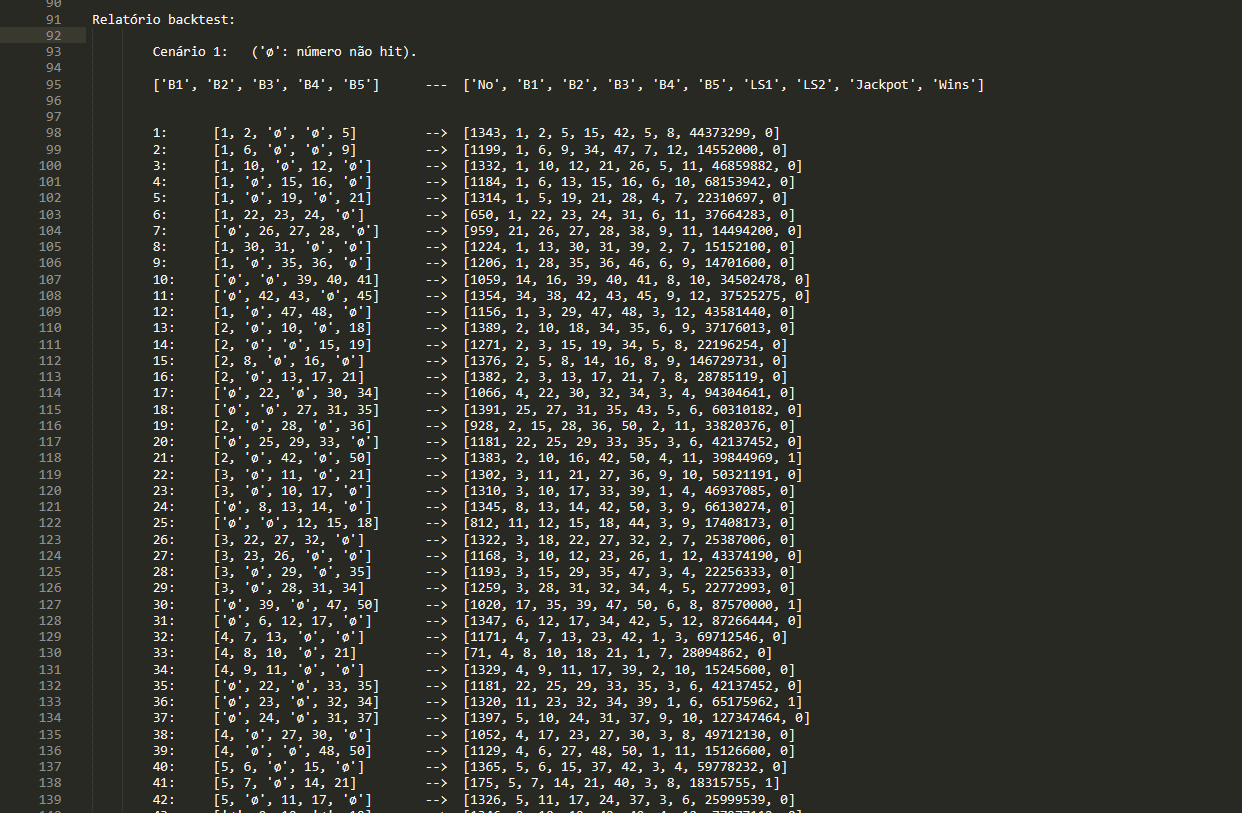


Segundo: Output do programa 9 (relatório).

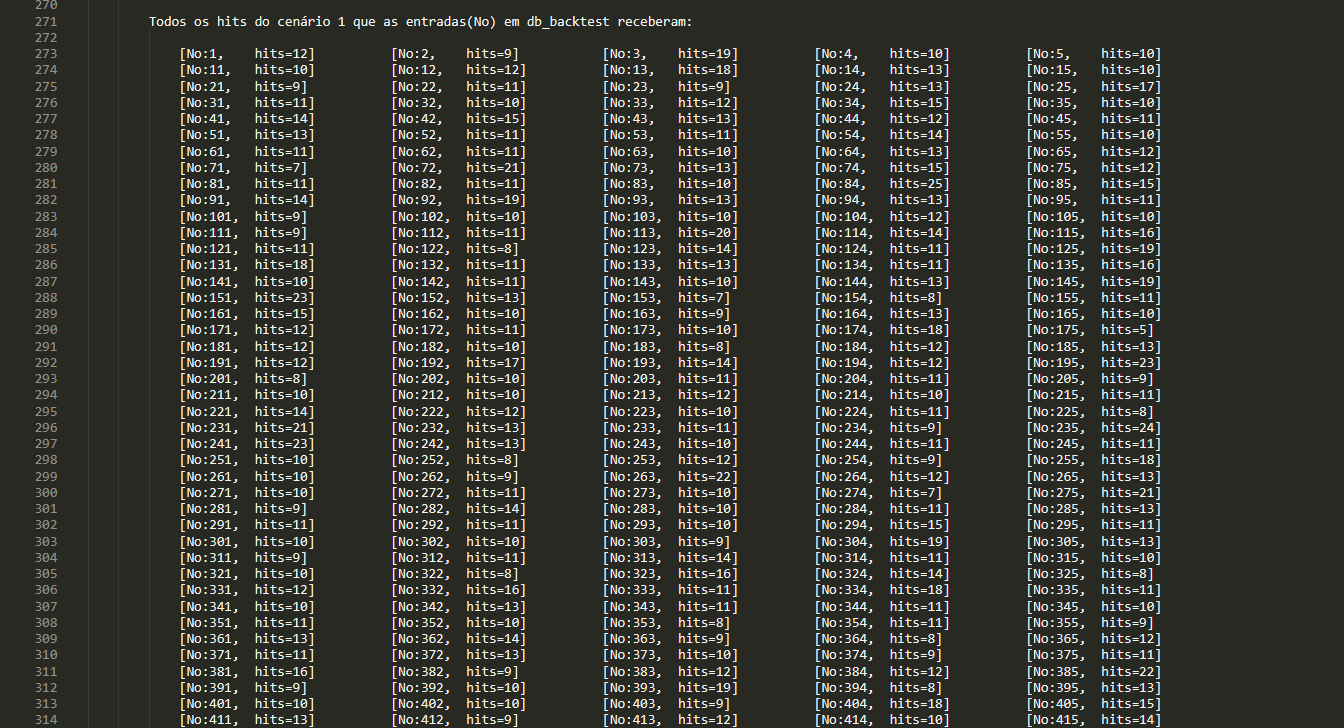
- Detalhe: o relatório gerado possui ~ 62.000 linhas. Para cada cenário, o relatório define três blocos: (1) a lista dos cartões apostados e, ao seu lado, o cartão da base euro\_concursoos.csv que teve mais acertos para o cartão apostado (i), ø significa que o número no cartão apostado i não teve um hit no cartão com mais hits da base .csv em relação a (i); (2) contém todos os cartões da base (representados por ‘No:‘ e seu número) e o número de cartões apostados que tiveram 2 hits ou mais; e (3) o total de cartões investidos no cenário X com a remuneração para as regras de contagem de cartões e remuneração nos programas 5, 6, 7 e 8 – basicamente, o que aconteceria se o apostador trocasse seus cartões seguindo as regras de um dos programas para obter a melhor remuneração.

-- Detalhe 2: como os cenários 1, 2 e 3 seguem a mesma lógica de output, vou mostrar apenas os blocos 1, 2 e 3 do cenário 1 para efeitos explicativos. Os resultados completos dos cenários estão salvos no arquivo txt: PRIME\_console.txt no folder console.

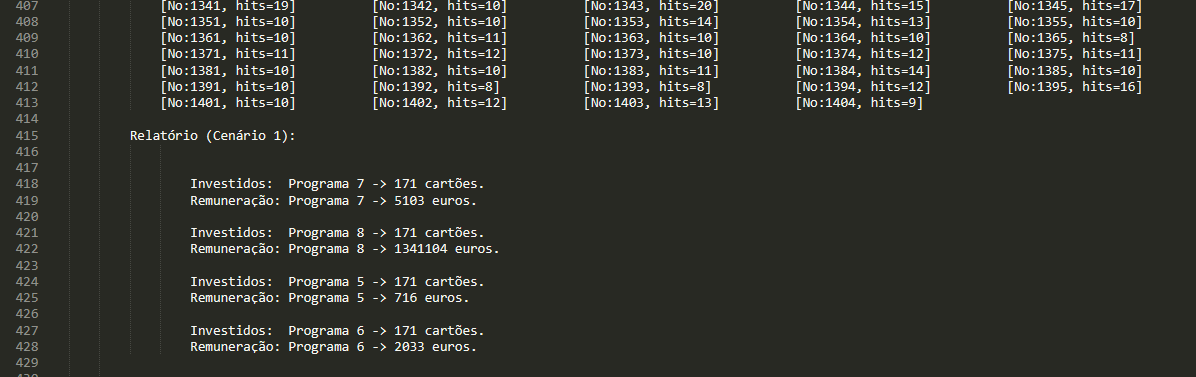
Bloco (1):



Bloco (2):

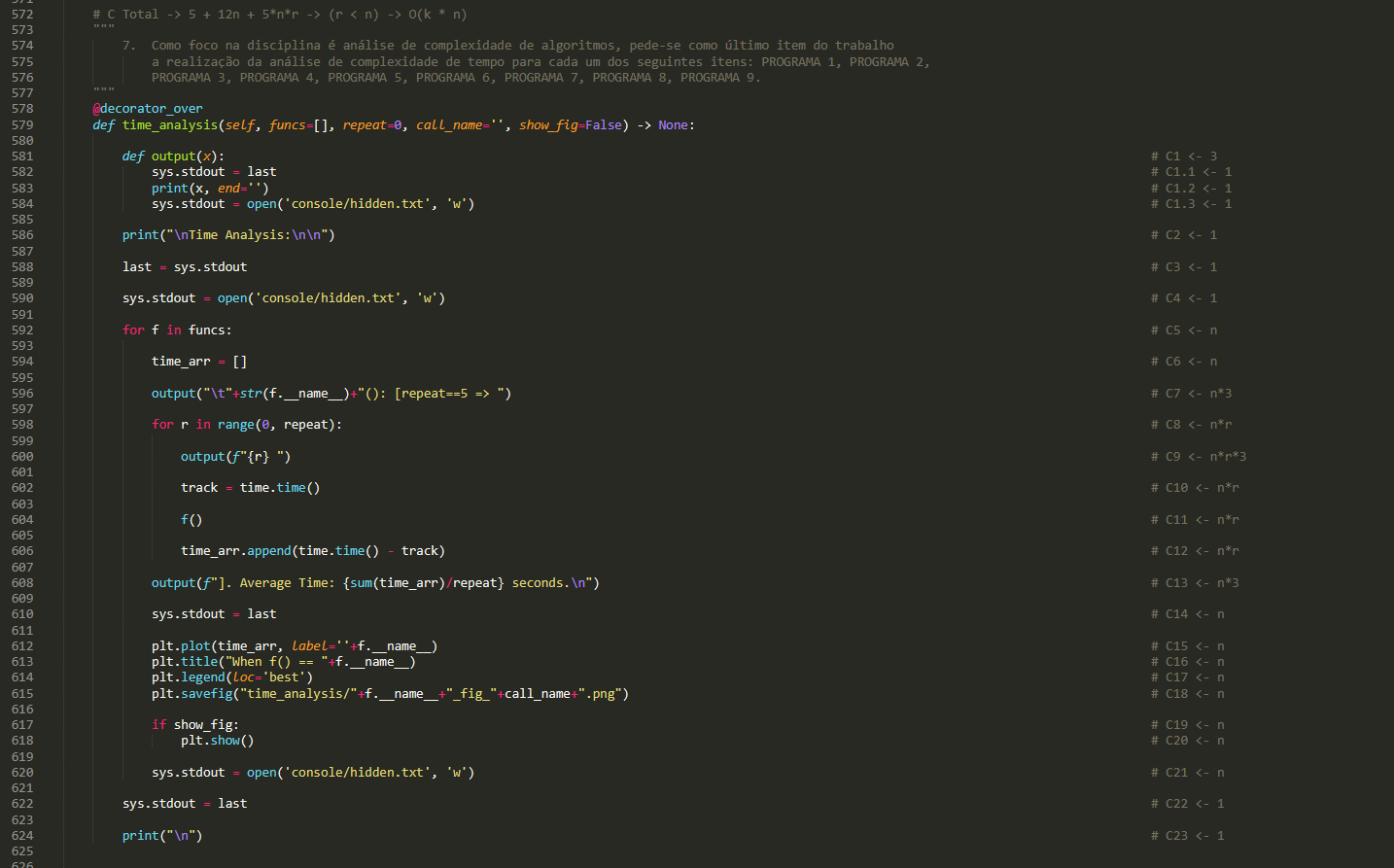


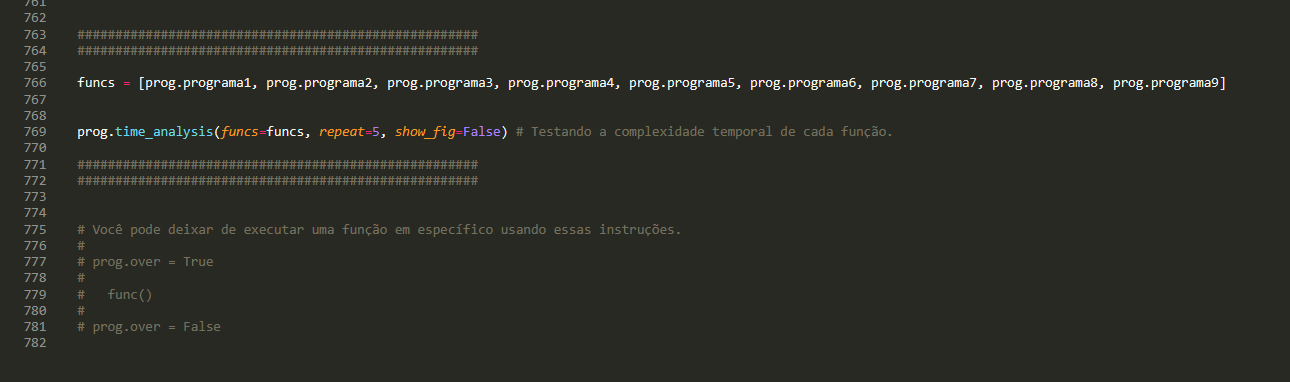
Bloco (3):



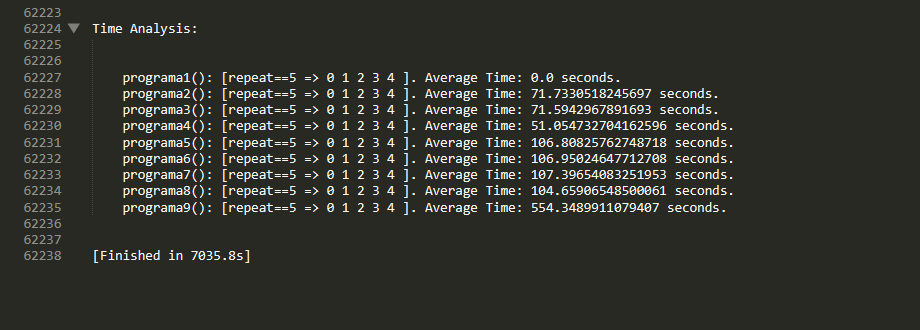
1. Como foco na disciplina é análise de complexidade de algoritmos, pede-se como último item do trabalho a realização da análise de complexidade de tempo para cada um dos seguintes itens: PROGRAMA 1, PROGRAMA 2, PROGRAMA 3, PROGRAMA 4, PROGRAMA 5, PROGRAMA 6, PROGRAMA 7, PROGRAMA 8, PROGRAMA 9.

Primeiro: o algoritmo que faz a análise temporal para todos os programas acima.





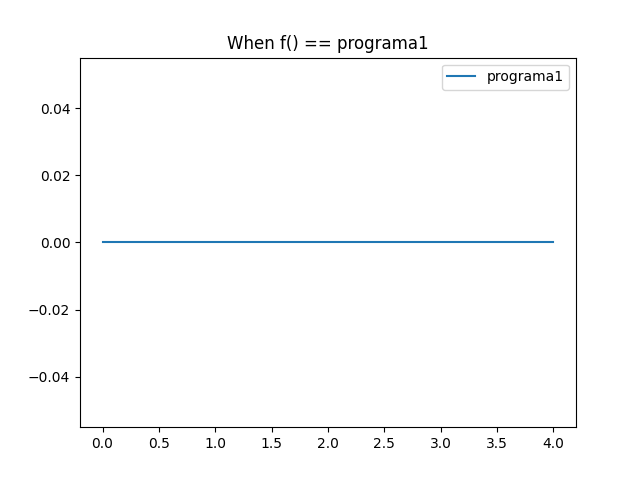
Segundo: o output do algoritmo.



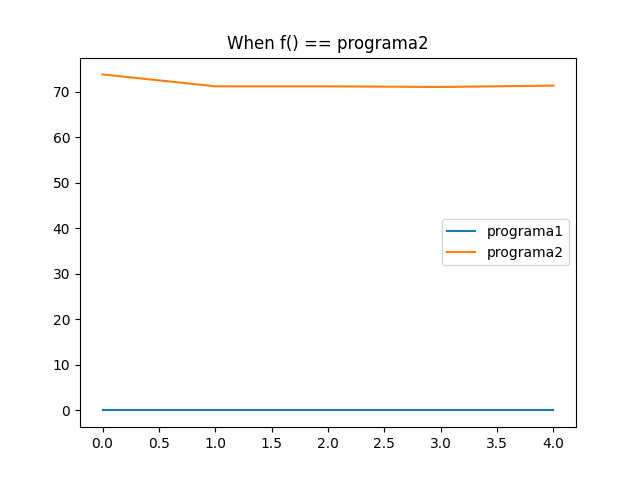
São ~ 62.000 linhas mesmo, haha.

Terceiro: os plots para cada programa de 1 a 9 de suas execuções.

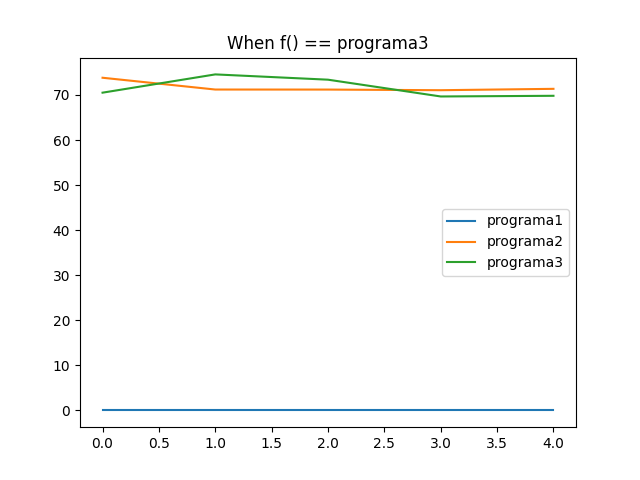
- Programa 1:



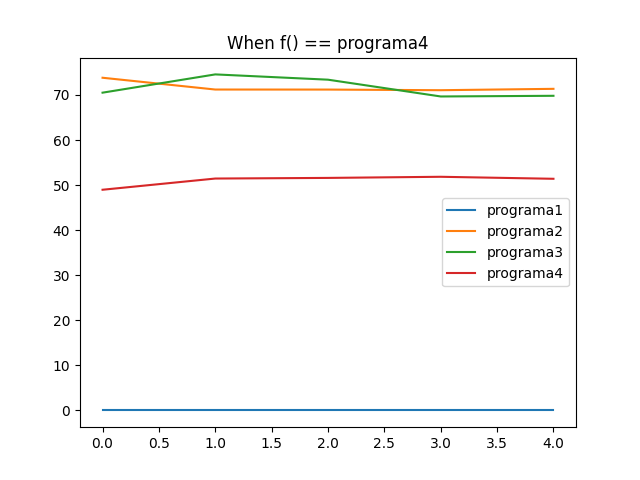
- Programa 2:



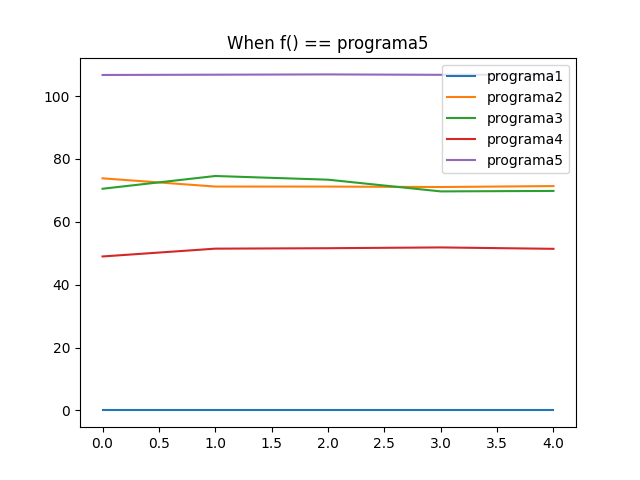
- Programa 3:



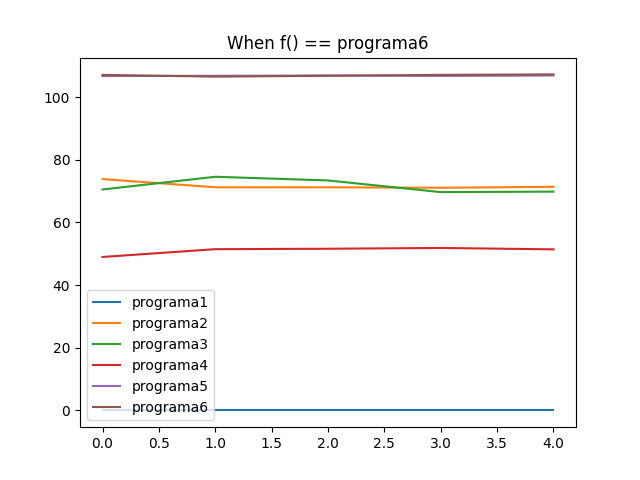
- Programa 4:



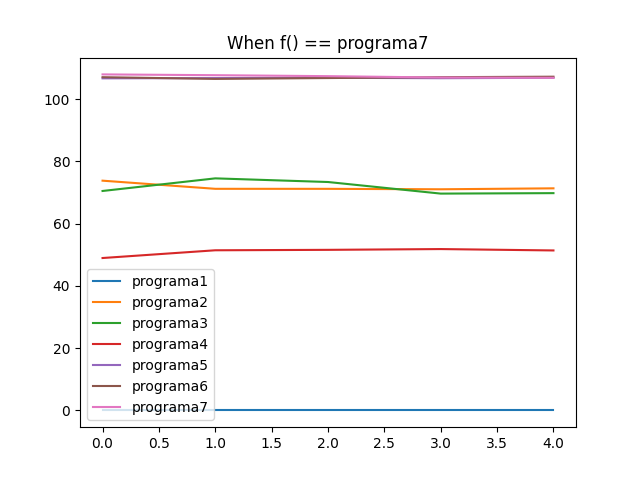
- Programa 5:



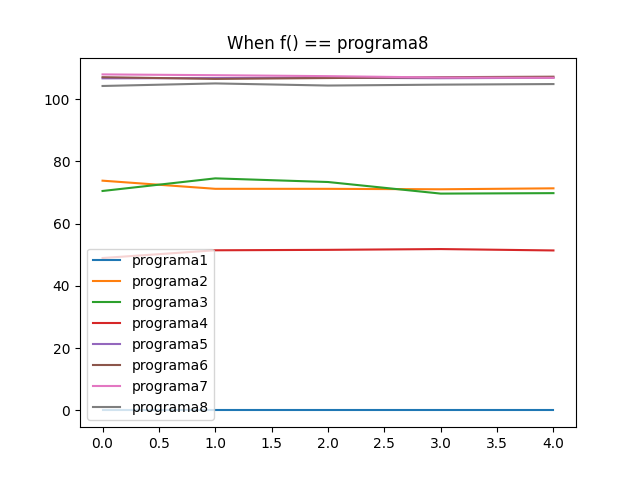
- Programa 6:



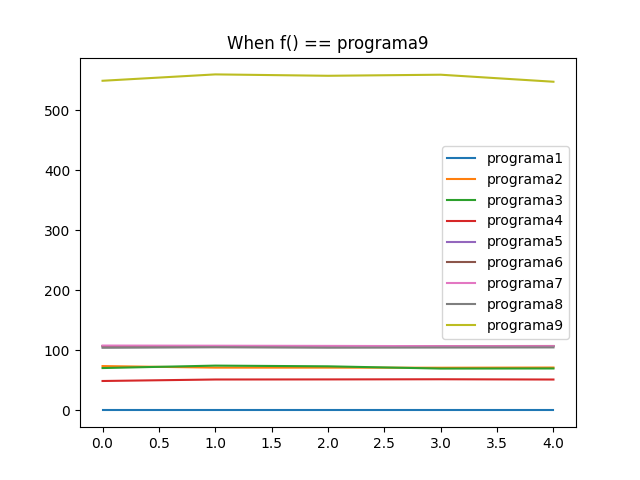
- Programa 7:



- Programa 8:



- Programa 9:



O programa leva, em média, 16 minutos para executar por completo, à exceção da função time\_analysis(), que executa cada programa 5 vezes, aumentando o tempo total para, aproximadamente, 1 hora e 40 minutos.

Fiz algumas funções e tricks para executar partes do código à sua escolha. Veja o arquivo.py, deixei tudo comentado, etapa a etapa.

Veja o vídeo da execução.

A tabela abaixo mostra uma parte do arquivo de teste (“euro\_concursos.csv”) com o histórico dos concursos. As colunas **B1, B2, B3, B4** e **B5** são os números sorteados em cada concurso. A coluna **Jackpot** é o valor do prêmios do concurso e a coluna **Wins** é o número de ganhadores. As colunas **LS1** e **LS2** são números de estrelas do “euro-millions”, não consideradas no processo de geração de combinações.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **B1** | **B2** | **B3** | **B4** | **B5** | **LS1** | **LS2** | **Jackpot** | **Wins** |
| 1404 | 4 | 5 | 39 | 46 | 48 | 7 | 10 | 27.914.329,00 | 0 |
| 1403 | 2 | 19 | 24 | 26 | 49 | 6 | 7 | 14.728.800,00 | 0 |
| 1402 | 6 | 12 | 22 | 29 | 33 | 6 | 11 | 182.028.000,00 | 1 |
| 1401 | 9 | 18 | 30 | 39 | 45 | 1 | 3 | 180.663.000,00 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | .... |
| 0004 | 4 | 7 | 33 | 37 | 39 | 1 | 5 | 13.870.849,00 | 1 |
| 0003 | 14 | 18 | 19 | 31 | 37 | 4 | 5 | 11.880.304,00 | 0 |
| 0002 | 7 | 13 | 39 | 47 | 50 | 2 | 5 | 10.111.500,00 | 0 |
| 0001 | 16 | 29 | 32 | 36 | 41 | 7 | 9 | 10.143.000,00 | 1 |