



INSTITUTO FEDERAL DE MINAS GERAIS

Bacharelado em Ciência da Computação

Disciplina: Matemática Discreta

Trabalho Prático 01

Professor: Diego Mello da Silva

Formiga-MG
15 de dezembro de 2013

Sumário

1	Introdução	1
2	Conjectura de Goldbach	1
3	Conjectura de Collatz	2
4	Especificação	2
4.1	Linguagem e ambiente de desenvolvimento	3
4.1.1	Artefatos (Entregáveis)	3
4.1.2	Compilação	3
4.2	Requisitos	4
4.2.1	Entrada de Dados	4
4.2.2	Estruturas de Dados	4
4.2.3	Cálculo de Números Primos até N	4
4.2.4	Conjectura de Goldbach	5
4.2.5	Conjectura de Collatz	5
4.2.6	Saída de Dados	5
4.2.7	Documentação de Código e Relatório	6
4.3	Barema de Correção	6
5	Relatório	6
6	Considerações Finais	7

1 Introdução

Este documento descreve a especificação do Trabalho Prático 01 da disciplina Matemática Discreta, e deve ser seguido de forma a contemplar os itens considerados na avaliação por parte do professor da disciplina. O trabalho é em grupo de até 3 alunos, cuja formação deve ser comunicada com antecedência para o professor. O trabalho prático tem o valor de 10 pontos.

O tema deste trabalho prático é a implementação de soluções computacionais para a busca de contra-exemplos para conjecturas matemáticas em aberto, e tem por objetivo (i) permitir que o aluno aprofunde em assuntos estudados na disciplina, buscando resolvê-los por meio de implementações computacionais; (ii) estimular o interesse pela disciplina de matemática discreta; (iii) estimular a prática de programação de computadores usando linguagem de programação de alto nível; (iv) promover o trabalho em grupo, o espírito de equipe, a divisão de tarefas e a liderança.

Neste trabalho serão abordadas duas conjecturas: a conjectura de Goldbach, e a conjectura de Collatz. Ambas as conjecturas estão em aberto até os dias de hoje, de forma que a dificuldade de demonstrá-las usando as técnicas tradicionais de prova têm levado cientistas e matemáticos a buscar por contra-exemplos que refutem tais conjecturas por provas exaustivas.

Devido às limitações de poder computacional dos computadores pessoais este trabalho irá restringir a declaração de ambas as conjecturas à um limite superior N que deve ser informado pelo usuário (ver seção de especificação). Assim, as declarações devem ser verificadas até o limite N informado.

2 Conjectura de Goldbach

A conjectura de Goldbach foi proposta pelo matemático Christian Goldbach em 1742, em uma carta ao famoso matemático Leonard Euler, que declara que **“todo número par maior ou igual a quatro é a soma de dois números primos”**. Sabe-se que pela definição de primalidade, um número primo é um número inteiro $p > 1$ tal que p não é divisível por nenhum inteiro além de 1 e p . Portanto, pela conjectura, qualquer número par n a partir de quatro pode ser escrito pela soma de dois primos p_1 e p_2 tais que $p_1 < n$ e $p_2 < n$.

Para ilustrar a declaração desta conjectura, seja a seguinte tabela, com os primeiros naturais pares após 4:

Número Par	Primo 1	Primo 2
4	2	2
6	3	3
8	3	5
10	3	7
12	5	7
14	7	7
16	3	13
18	5	13
20	7	13
22	17	5
\vdots	\vdots	\vdots

Embora a conjectura pareça verdadeira para qualquer número maior ou igual a 4, não existe prova dedutiva que demonstre tal verdade, e sim apenas a observação do fato para números já verificados por computador. A subseção 4.2.4 detalhará o que deve ser implementado pelo grupo neste trabalho prático.

3 Conjectura de Collatz

Também conhecida pela variante denominada ‘Problema $3n + 1$ ’, a conjectura de Collatz foi declarada pelo matemático Lothar Collatz em 1937. Na conjectura de Collatz, **“dado um número natural n que segue a seguinte regra de formação:**

$$f(n) = \begin{cases} n/2, & \text{se } n \text{ for par} \\ 3n + 1, & \text{se } n \text{ for ímpar} \end{cases}$$

então a sequência gerada por esta regra sempre encerra em 1”. Chama-se *órbita do número n* a sequência gerada de n até 1 segundo esta regra de formação.

Para ilustrar como isso ocorre, seja a Figura 1 que mostra a sequência gerada a partir do número $n = 3$ que resulta em 3, 10, 5, 16, 8, 4, 2, 1, e a Figura 2 para $n = 9$ que resulta em 9, 28, 14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, ambos exemplos retirados do artigo *Formulações Alternativas para a Conjectura de Collatz*, de Filho A. C. S.

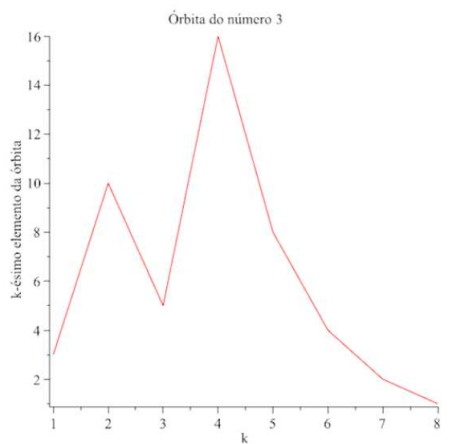


Figura 1: Órbita para $n = 3$

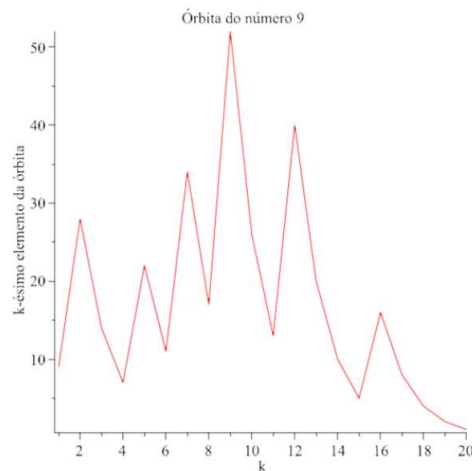


Figura 2: Órbita para $n = 9$

4 Especificação

Esta seção apresenta a especificação do trabalho. Todas as diretivas devem ser seguidas pelo grupo sob pena de perda de pontos ou invalidação. Caso o grupo pretenda fazer algo diferente do especificado deverá pedir permissão por escrito ao professor da disciplina antes de aplicar tais modificações.

4.1 Linguagem e ambiente de desenvolvimento

4.1.1 Artefatos (Entregáveis)

Todo o trabalho deverá ser implementado na linguagem C ou Free Pascal, e compilada nos seguintes compiladores `fpc` e `gcc`, ambos disponíveis nas distribuições Linux. Assume-se que o código foi compilado e testado no ambiente Linux Ubuntu ou similar, pois este será o ambiente de correção pelo professor.

O grupo deverá entregar apenas três arquivos para o professor, compactados em arquivo no formato `.zip`: código fonte da aplicação, denominado `conjectura.pas` ou `conjectura.c`; o relatório do projeto (formato `.pdf`) segundo o modelo descrito na seção 5, e um arquivo `grupo.txt` contendo o nome e matrícula dos membros do grupo. O arquivo deverá ser enviado **até a meia noite** da data de entrega para o email `diego.silva@ifmg.edu.br`.

O arquivo de código fonte deverá estar comentado e devidamente modularizado, isto é, é preciso dividir a implementação em funções com **pelo menos** o cabeçalho das funções comentadas, descrevendo o que cada uma delas faz. Lembre-se que a documentação é importante para compreensão do código por terceiros, entre eles o professor.

4.1.2 Compilação

Para compilar programas escritos em Pascal no `fpc`, abra um terminal e acesse a pasta onde encontra-se o arquivo com o código fonte. Em seguida use o seguinte comando via terminal do Linux:

```
user@machine$ fpc conjectura.pas -o conjectura.bin
```

Para compilar programas escritos em C no `gcc`, abra um terminal e acesse a pasta onde encontra-se o arquivo com o código fonte. Em seguida use o seguinte comando via terminal do Linux:

```
user@machine$ gcc conjectura.c -o conjectura.bin
```

Em ambos os casos, se o programa não contiver erros de sintaxe, então o compilador `fpc/gcc` irá gerar o arquivo de saída denominado `conjectura.bin`, que é o programa executável. Para executá-lo, digite no terminal:

```
user@machine$ ./conjectura.bin
```

Importante: Códigos fonte incompletos, com erros de sintaxe ou execução valerão zero, portanto procure testar adequadamente sua implementação antes de enviar a versão final. Todos os laboratórios do campus possuem as ferramentas necessárias disponíveis. Em caso de dúvidas/dificuldades, procurar o professor ou enviar email para `diego.silva@ifmg.edu.br` com no máximo 1 semana de antecedência à data de entrega.

4.2 Requisitos

A implementação deste trabalho será dividida em requisitos de *software* que devem ser implementados conforme especificado a seguir. É esperado que cada um dos requisitos de *software* abaixo seja traduzido em uma ou mais funções na linguagem C ou Pascal, garantindo a modularização de código e a divisão de um problema em sub-problemas menores.

4.2.1 Entrada de Dados

A entrada de dados do aplicativo `conjectura.bin` será feita via console, ou seja, em linha de comando. Uma vez que a aplicação seja executada deverá imprimir na tela do usuário as seguintes mensagens e perguntas:

```
Verificador de Conjectura
===== == =====
```

```
Verificar Goldbach (G) ou Collatz (C): _
```

```
Limite Superior                               : _
```

O usuário deverá responder primeiro se testará a conjectura de Goldbach (caso tenha inserido o caracter ‘G’) ou Collatz (caso tenha inserido o caracter ‘C’). Após informar qual é a conjectura em teste, o usuário deverá informar qual é o limite superior da verificação, ou seja, o valor de N para o qual a conjectura será testada.

Ambas as informações devem ser guardadas na memória como parâmetro para o método.

4.2.2 Estruturas de Dados

Este requisito solicita que o grupo deixe explícito quais foram as estruturas de dados (variáveis primitivas, compostas e/ou definidas pelo usuário) usadas para guardar os parâmetros e resultados intermediários (como os números primos no caso da conjectura de Goldbach) e finais do verificador de conjecturas. Sugere-se comentar cada uma delas por motivos de clareza e pontuação.

Outra observação que o grupo deverá fazer está relacionada com a representação numérica. Ambos as conjecturas trabalham no domínio dos números naturais (ou inteiros positivos). No entanto, as linguagens de programação não conseguem representar números inteiros excessivamente grandes. O grupo deverá pesquisar dentre os tipos de dados disponíveis qual melhor se adequa para os cálculos de ambas as verificações, justificando neste requisito o porquê da sua escolha, ou limitando o valor máximo de N em um número onde as operações efetuadas sejam garantidamente suportadas.

4.2.3 Cálculo de Números Primos até N

Em particular a conjectura de Goldbach utiliza números primos p calculados entre $4 \leq p \leq N$, de forma a verificar se cada número par n maior ou igual a 4 pode ser formado pela soma de dois primos.

Para esta implementação sugere-se o estudo e implementação do algoritmo conhecido como Crivo de Eratóstenes. No entanto, o grupo poderá formular seu próprio algoritmo de cálculo de números primos. Importante ressaltar que os resultados calculados devem ser armazenados para a verificação da conjectura.

4.2.4 Conjectura de Goldbach

Neste trabalho o grupo deverá implementar um verificador para esta conjectura para números inteiros pares presentes no intervalo $4 \leq n \leq N$, onde N é o limite superior informado pelo usuário.

O grupo deverá ter em mente que pode-se testar a soma de dois números primos pré-calculados, dois a dois, em busca de uma soma que resulte no valor de um dado inteiro par n , onde $4 \leq n \leq N$. Caso a soma resulte em n , então a conjectura está verificada para n e precisa ser verificado agora para o próximo inteiro par $(n + 2)$. Proceder até que todos os inteiros contidos no intervalo sejam verificados.

4.2.5 Conjectura de Collatz

Este requisito solicita a implementação de um método computacional para verificar a conjectura de Collatz, que gera sequência de números segundo a regra de formação apresentada na seção 3 e, de acordo com este matemático, sempre se encerra em 1.

Esta verificação deve ocorrer para números na faixa $1 < n \leq N$, onde N é o limite superior informado pelo usuário.

4.2.6 Saída de Dados

A saída de dados será feita em console, exibindo resultados computados de maneira diferente para cada conjectura testada.

Se a conjectura testada foi a conjectura de Goldbach, então deve-se imprimir a cada número par n verificado qual foi a combinação de primos cuja soma resultou no inteiro par em questão, um por linha. Exemplo:

```
Goldbach 4 = 2 + 2 (Ok)
Goldbach 6 = 3 + 3 (Ok)
Goldbach 8 = 3 + 5 (Ok)
...
```

Se a conjectura testada foi a conjectura de Collatz, então deve-se imprimir a sequência de números gerados a partir de n que avançou segundo a regra descrita na seção 3 (isto é, de acordo com a regra de formação descrita por Collatz), no seguinte formato de exemplo:

```
Collatz 1 (Ok)
Collatz 2 -> 1 (Ok)
Collatz 3 -> 10 -> 5 -> 16 -> 8 -> 4 -> 2 -> 1 (Ok)
Collatz 4 -> 2 -> 1 (Ok)
...
```

Mensagens de erro e outras mensagens que sejam pertinentes à aplicação devem ser elaboradas e descritas pelo grupo a parte, no relatório de implementação deste verificador de conjecturas.

4.2.7 Documentação de Código e Relatório

Embora este requisito não se trate de um requisito de *software*, ele deve ser considerado como fundamental para a correção do trabalho.

Neste requisito solicita-se que o código fonte seja documentado através de comentários de código. Os comentários não deve ser totalmente ausentes nem totalmente massivos. Uma boa prática de documentação por comentário é documentar os cabeçalhos de cada função, assim como as linhas mais importantes da função.

O relatório de implementação também é um artefato esperado para correção, e deve conter os detalhes, dificuldades, decisões de projeto e outros fatos ocorridos ao longo do trabalho. A especificação sobre o formato e conteúdo das seções do relatório é dado a seguir, na seção 5. A versão final do mesmo a ser enviada ao professor deverá estar, **obrigatoriamente**, no formato pdf.

4.3 Barema de Correção

Conforme mencionado, o trabalho prático tem valor de 10 pontos. A correção seguirá o barema apresentado a seguir, que descreve os requisitos e pontos associados.

Requisito	Pontos
Entrada de Dados	1 pto
Estruturas de Dados	1 pto
Cálculo dos Números Primos até N	1 pto
Conjectura de Goldbach	2 ptos
Conjectura de Collatz	2 ptos
Saída de Dados	1 pto
Documentação de Código e Relatório	2 ptos
TOTAL	10 ptos

A pontuação do grupo será feita mediante a entrega integral dos requisitos acima, verificados no código fonte e no teste em ambiente Linux. Obviamente, alguns requisitos impedem a implementação ou execução dos demais (por exemplo, não entregar o requisito ‘Cálculo dos Números Primos até N ’ impede que o requisito ‘Conjectura de Goldbach’ seja também entregue e avaliado).

5 Relatório

Juntamente com os demais artefatos descritos neste documento é necessário que o grupo entregue um breve relatório de implementação, que deverá conter as seguintes seções:

1. **Introdução**, que explica o problema que está sendo resolvido com uma breve revisão de literatura sobre o problema;
2. **Grupo**, que apresenta os membros do grupo (nome e matrícula), o líder do grupo e a divisão de trabalho realizada (atenção a este item, pois cada membro poderá ser arguido e pontuado segundo sua atuação e carga de trabalho);

3. **Ambiente de desenvolvimento**, que descreve o computador, sistema operacional, detalhes sobre a CPU e memória, compiladores, IDEs e ferramentas usadas para construir a implementação, assim como instruções para compilação e execução da aplicação final;
4. **Estrutura de Dados**, onde os tipos de dados primitivos e definidos pelo usuário, vetores e afins deverão ser apresentados e justificados;
5. **Algoritmos**, onde o grupo deverá apresentar em pseudo-código o esqueleto do código fonte com as operações mais importantes da implementação. Este pseudo-código não é a implementação propriamente dita, e sim uma descrição das principais operações realizadas em um nível de entendimento suficiente para um não-programador entender sua solução.
6. **Conclusões**, onde o grupo deverá apresentar seu parecer de forma técnica, profissional e impessoal sobre os objetivos do trabalho, dificuldades encontradas, aprendizado, limitações encontradas pela tecnologia usada, decisões de projeto importantes tomadas pelos membros do grupo, entre outros.

6 Considerações Finais

O trabalho prático especificado neste documento procura desenvolver habilidades técnicas e interpessoais entre os membros da equipe de maneira saudável. Para tal, é fundamental que o trabalho seja feito de maneira ética e profissional. Desta forma algumas considerações finais devem ser feitas:

- Este é um trabalho de programação, portanto aqueles que possuem dificuldades devem estudar o assunto, procurando livros, tutoriais e outros materiais que complementem sua deficiência em programação por conta própria;
- Trabalhos plagiados ou feitos por terceiros valerão zero;
- Todos os membros da equipe deverão participar efetivamente do trabalho de forma a aumentar suas perícias em matemática e programação;
- Não deve-se ‘carregar peso morto’: jamais inclua um membro na equipe na entrega do trabalho se ele não contribuir em nada para o mesmo;
- Caso o professor julgue necessário poderá fazer uma arguição dos membros de um ou mais grupos;
- Trabalhos entregues fora do prazo serão desconsiderados;
- Dúvidas sobre o trabalho devem ser tiradas com no máximo 1 semana do prazo de entrega.