Relatório do Projeto Prático 01 de Fundamentos de Programação 1

Rafael L. Marinheiro, Ruanitto R. Docini

Universidade Federal Tecnológica do Paraná (UTFPR) – Curitiba, PR – Brasil

rafaellmarinheiro42@gmail.com, rdocini@alunos.utfpr.edu.br

Abstract. This meta-paper is an explanation of the first project in the subject “Fundamentos de Programação 1” from the course of Information Systems, made in pairs.

Resumo. Este artigo é a explicação do primeiro projeto da matéria de “Fundamentos de Programação 1” do curso de Sistemas de Informação, feito em duplas.

1. Calendário

1.1 Apresentação do Problema

Nesse primeiro problema do projeto o objetivo era o desenvolvimento de um programa que gerasse um calendário de um determinado mês e ano, através de uma leitura feita do teclado e exibindo a formatação de um calendário tradicional na tela.

1.2 Resolução

O primeiro passo a ser dado pela dupla foi a questão da formatação da tela. Utilizamos o exemplo dado como base, e trabalhamos com "printf()", dando espaços, pulando linhas e escrevendo os dias de domingo a sábado. Depois, foi feito uma função para identificar o nome do mês dado e a quantidade de dias que ele possui. Caso o mês fosse fevereiro, é preciso verificar se o ano é bissexto, pois em anos bissextos fevereiro tem 29 dias. Usamos a congruência de Zeller, que pode ser observada na imagem abaixo:

$$h = \left(q + \left\lfloor \frac{13(m+1)}{5} \right\rfloor + K + \left\lfloor \frac{K}{4} \right\rfloor + \left\lfloor \frac{J}{4} \right\rfloor + 5J\right) \bmod 7,$$

Figura 1. Fórmula da Congruência de Zeller

A variável "h" resultará em um número de 0 à 6, sendo cada um dia da semana, 0 sábado, 1 domingo e assim por diante. "m" é o mês dado, "k" é o ano módulo de 100 e "j" é o ano divido por 100. Convertemos o valor inicial para domingo, pois nosso calendário começa no domingo. Depois, deixamos em branco os dias de outros meses na primeira semana e preenchemos o resto do mês com os dias nos locais corretos.

2. Uma Conjectura Falsa de Euler

2.1 Apresentação do Problema

Este outro problema pedia para que fosse feita uma solução computacional para o a conjectura de Euler no caso da potência n = 5. A conjectura diz que um n número de termos elevados a x, onde x = n+1, não pode ser igual a um único número elevado a x, como no exemplo abaixo:

a^5+b^5+c^5+d^5 = e^5

2.2 Resolução

Nossa primeira tentativa foi fazer uma força bruta, testando todas as possibilidades, para depois poder aprimorá-la. Fizemos as 5 variáveis e comparamos umas com as outras de todas as formas. Isso resultou em um programa que demorava em torno de 8 segundos para achar a solução. Discutindo com alguns colegas e olhando alguns códigos-fonte na internet, percebemos que seria melhor utilizar um vetor para armazenar todas as soluções em vez de fazê-las a todo momento. Isso diminuiu o tempo de execução para aproximadamente 6 segundos. Como solução final, alteramos o código para que em vez de chegar em um teto, definido por nós, ele faria um novo teto todo início da repetição, começando no maior até chegar ao 0, e caso ele não ache o resultado, ele incrementa o teto em 1. Então na primeira passagem, iniciamos as variáveis como: a = 4, b = 3, c = 2 e d = 1. Na segunda passagem, a = 5, b = 4, c = 3, d = 2 e assim por diante, fazendo o seguinte algorítimo: a = n, b = a-1, c = b-1, d=c-1. Assim que o programa acha uma solução, ele iguala todas as variáveis a um número negativo (no caso, o -1) pois a condição de parada das repetições é que a variável seja menor ou igual a 0.

3. Números Romanos

3.1 Apresentação do Problema

No último problema do projeto a proposta era escrever um programa que apresentasse na tela o valor arábico de um número romano, lido do teclado pelo usuário.

3.2 Resolução

A primeira coisa a ser feita foi a criação de uma função que converteria um número romano simples (um algarismo) para o seu equivalente em arábico, de acordo com os valores abaixo:

I = 1, V = 5, X = 10, L = 50, C = 100, D = 500, M = 1000

Depois fizemos uma estrutura de repetição que lê os algarismos romanos e a cada vez que um novo é digitado, ele já faz as conversões necessárias. Primeiro verificamos se o algarismo anterior é menor do que o atual. Caso isso seja verdade, precisamos subtrair o anterior do atual e do total também. Depois adicionamos o atual no total, e, por fim, atribuímos o valor atual a uma variável auxiliar, que será o “valor anterior”. Repetimos esse processo até o usuário digitar ENTER e imprimimos o resultado.

4. Referências

Autor Anônimo “Zellers Congruence” (Atualizado em 31 de agosto de 2017) https://en.wikipedia.org/wiki/Zeller%27s_congruence

Autores Anônimos "Euler's Sum of Powers Conjecture" (Atualizado em 27 de agosto de 2017) https://rosettacode.org/wiki/Euler%27s_sum_of_powers_conjecture