Aula 2 - Lista 1.2 de exercícios de Programação e Estruturas de Dados II

Problema 1.2.1
Fazer um algoritmo que:
Leia um número indeterminado de linhas contendo cada uma um número inteiro. A última linha, que não entrará nos cálculos, contém o valor do número igual a zero. Calcule e escreva, para cada número válido, se este é ou não primo.

Problema 1.2.2
Fazer um algoritmo que:
Leia um número indeterminado de linhas contendo cada uma um número inteiro. A última linha, que não entrará nos cálculos, contém o valor do número igual a zero. Calcule e escreva, para cada número válido, se este é ou não perfeito. Número perfeito é aquele cuja soma de seus divisores, exceto ele próprio é igual ao número. Exemplo: 6 é perfeito porque 1 + 2 + 3 = 6.

Problema 1.2.3
Fazer um algoritmo que:
Leia um número indeterminado de linhas contendo cada uma um número inteiro. A última linha, que não entrará nos cálculos, contém o valor do número igual a zero. Calcule e escreva, para cada número válido, se este é ou não um quadrado perfeito. Um número será quadrado perfeito quando respeitar a regra de formação: n^2= a. Nessa regra, n é qualquer número inteiro positivo e a é o número quadrado perfeito. Ex.:

$1^2 = 1 . 1 = 1$ $2^2 = 2. 2 = 4$ $3^2 = 3 . 3 = 9$ $4^2 = 4 . 4 = 16$ $5^2 = 5 . 5 = 25$	$6^2 = 6 . 6 = 36$ $7^2 = 7 . 7 = 49$ $8^2 = 8 . 8 = 64$ $9^2 = 9 . 9 = 81$ $10^2 = 10 . 10 = 100$

Problema 1.2.4
Considerando o algoritmo anterior, verifique se o número adquirido obedece às seguintes regras:

Primeira Regra: Somente o número quadrado perfeito possui raiz quadrada exata. Exemplos - Veja o cálculo da raiz quadrada dos números a seguir:

$$\sqrt[2]{4}=\sqrt[2]{2.2}=\sqrt[2]{2^2}=2, \text{ logo: } \sqrt[2]{4}=2$$
$$\sqrt[2]{9}=\sqrt[2]{3.3}=\sqrt[2]{3^2}=3, \text{ logo: } \sqrt[2]{9}=3$$
$$\sqrt[2]{25}=\sqrt[2]{5.5}=\sqrt[2]{5^2}=5, \text{ logo: } \sqrt[2]{25}=5$$

Segunda Regra: Quando o número é quadrado perfeito, ele não possui como último algarismo os seguintes números: 2, 3, 7 e 8.

Terceira Regra: Todo número quadrado perfeito que for par possuirá raiz quadrada par. Lembre-se de que um número é considerado par quando for dividido por dois e resultar em um número inteiro. Exemplos - verifique se os números 4, 9 e 16 são pares e calcule a raiz quadrada deles:

4 : 2 = 2 → Temos que 4 é um número par; 9 : 2 = 4,5 → O número 9 não é par; 16 : 2 = 8 → O número quadrado perfeito 16 é par.

$$\sqrt[2]{4}=2$$
$$\sqrt[2]{16}=4$$

Quarta Regra: Um número par será quadrado perfeito se, ao ser dividido por 4, resultar em um número inteiro.

Quinta Regra: Todo número quadrado perfeito que é ímpar possui raiz quadrada ímpar. Um número será ímpar quando ele for dividido por dois e resultar em um número que não é inteiro, ou seja, um número decimal. Exemplos - Considere os números 100 e 121. Verifique qual é ímpar e calcule a sua raiz quadrada.

100 : 2 = 50 → 100 é par; 121 : 2 = 60,5 → 121 é ímpar.

A raiz quadrada de 121 é 11. Sendo assim, a quinta regra é válida, pois número quadrado perfeito ímpar possui raiz quadrada ímpar.

Sexta Regra: Ao dividir um número quadrado perfeito ímpar por oito, o resto sempre será o número 1. Exemplos - Verifique se os números 9 e 25 deixam resto 1 ao serem divididos por 8:

<table>
<tr><td>9 | 8
8 1
1</td><td>25 | 8
24 3
1</td></tr>
</table>

Observando as divisões acima, verificamos que a sexta regra é válida para os números que são ímpares e quadrados perfeitos.

Problema 1.2.5
Triângulo de Pascal é um triângulo aritmético infinito onde são dispostos os coeficientes das expansões binominais. Os números que compõem o triângulo apresentam diversas propriedades e relações. **Coeficiente Binomial**: os números que compõem o triângulo de Pascal são chamados de números binomiais ou coeficientes binomiais. Um número binomial é representado por:

$$\binom{n}{p}$$

Com *n* e *p* números naturais e n ≥ p. O número *n* é denominado numerador e o *p* denominador.

O número binomial é calculado a partir da relação:

$$\binom{n}{p} = C_{n,p} = \frac{n!}{p!(n-p)!}$$

Sendo,

$C_{n,p}$: combinação simples de n elementos tomados p a p
n!: fatorial de n, ou seja, n.(n - 1).(n - 2)...3.2.1
p!: fatorial de p, ou seja, p.(p - 1).(p - 2)...3.2.1

Construção do Triângulo

O triângulo de Pascal é construído colocando-se os números binomiais de mesmo numerador na mesma linha e os coeficientes de mesmo denominador na mesma coluna. Assim, temos:

<table>
<tr><td>linha 0 $\binom{0}{0}$
linha 1 $\binom{1}{0}\binom{1}{1}$
linha 2 $\binom{2}{0}\binom{2}{1}\binom{2}{2}$
linha 3 $\binom{3}{0}\binom{3}{1}\binom{3}{2}\binom{3}{3}$
linha 4 $\binom{4}{0}\binom{4}{1}\binom{4}{2}\binom{4}{3}\binom{4}{4}$
⋮
linha n $\binom{n}{0}\binom{n}{1}\binom{n}{2}\binom{n}{3}\binom{n}{4}\cdots\binom{n}{n}$</td><td>1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
...</td></tr>
</table>

Ao calcular os valores dos coeficientes, encontramos a seguinte representação do triângulo de Pascal (acima, à direita).

Propriedades

1.ª) **Todas as linhas têm o número 1 como seu primeiro e último elemento**.

De fato, o primeiro elemento de todas as linhas é calculado por:

$$\binom{n}{0} = \frac{n!}{0!(n-0)!} = \frac{n!}{0!n!} = 1 \qquad (considerando\ 0! = 1)$$

e o último elemento de todas as linhas é calculado por:

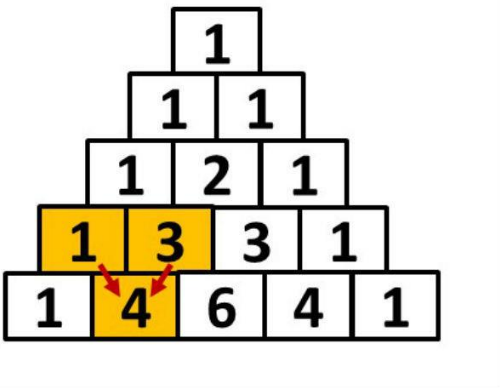
$$\binom{n}{n} = \frac{n!}{n!} = 1$$

2.ª) **O restante dos números de uma linha é formado pela adição dos dois números mais próximos da linha acima.**

Essa propriedade é chamada de Relação de Stifel e é expressa por:

$$\binom{n-1}{p-1} + \binom{n-1}{p} = \binom{n}{p}$$

É possível verificar a relação de Stifel diretamente no triângulo de Pascal, porque a partir da segunda linha, cada elemento é igual à soma do elemento acima com o seu anterior.



3.ª) **Os elementos de uma mesma linha equidistantes dos extremos têm valores iguais**.

$$\binom{n}{p} = \binom{n}{n-p} \rightarrow p + (n-p) = n \quad (números\ binomiais\ complementares)$$

Exemplos

a) $\binom{4}{1} = \binom{4}{3} \rightarrow 1 + 3 = 4$

b) $\binom{6}{2} = \binom{6}{4} \rightarrow 2 + 4 = 6$

4.ª) **A soma dos elementos de uma linha de numerador (n) será igual a 2^n**.

Observe a tabela:

Linha	Representação dos números no triângulo	Soma dos números	Resultado da soma na forma de potência
0	1	1	2^0
1	1 1	1+1	2^1
2	1 2 1	1+2+1	2^2
3	1 3 3 1	1+3+3+1	2^3
4	1 4 6 4 1	1+4+6+4+1	2^4
5	1 5 10 10 5 1	1+5+10+10+5+1	2^5

Implemente um código que gere o Triângulo de Pascal para n linhas e que permita testar a validade de um dado Triângulo através da validação de suas propriedades.

Problema 1.2.6 (OBI2000)

Quermesse

Os alunos do último ano resolveram organizar uma quermesse para arrecadar fundos para a festa de formatura. A festa prometia ser um sucesso, pois o pai de um dos formandos, Teófilo, dono de uma loja de informática, decidiu doar um computador para ser sorteado entre os que comparecessem. Os alunos prepararam barracas de quentão, pipoca, doces, ensaiaram a quadrilha e colocaram à venda ingressos numerados sequencialmente a partir de 1. O número do ingresso serviria para o sorteio do computador. Ficou acertado que Teófilo decidiria o método de sorteio; em princípio o sorteio seria, claro, computadorizado. O local escolhido para a festa foi o ginásio da escola. A entrada dos participantes foi pela porta principal, que possui uma roleta, onde passa uma pessoa por vez. Na entrada, um funcionário inseriu, em uma lista no computador da escola, o número do ingresso, na ordem de chegada dos participantes. Depois da entrada de todos os participantes, Teófilo começou a trabalhar no computador para preparar o sorteio. Verificando a lista de presentes, notou uma característica notável: havia apenas um caso, em toda a lista, em que o participante que possuía o ingresso numerado com i, havia sido a i-ésima pessoa a entrar no ginásio. Teófilo ficou tão encantado com a coincidência que decidiu que o sorteio não seria necessário: esta pessoa seria o ganhador do computador.

1. Tarefa) Conhecendo a lista de participantes, por ordem de chegada, sua tarefa é determinar o número do ingresso premiado, sabendo que o ganhador é o único participante que tem o número do ingresso igual à sua posição de entrada na festa.

2. Entrada) A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém um número inteiro positivo N que indica o número de participantes da festa. A linha seguinte contém a sequência, em ordem de entrada, dos N ingressos das pessoas que participaram da festa. O final da entrada é indicado quando N = 0. Para cada conjunto de teste da entrada haverá um único ganhador.

Exemplo de Entrada

4

4 5 3 1

10

9 8 7 6 1 4 3 2 12 10

0

3. Saída) Para cada conjunto de teste da entrada seu programa deve produzir três linhas. A primeira linha identifica o conjunto de teste, no formato "Teste n", onde n é numerado a partir de 1. A segunda 5 linha deve conter o número do ingresso do ganhador, conforme determinado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente. Exemplo de Saída

Teste 1

3

Teste 2

10

(esta saída corresponde ao exemplo de entrada acima)

Problema 1.2.7 (OBI2000)

Bits Trocados

As Ilhas Weblands formam um reino independente nos mares do Pacífico. Como é um reino recente, a sociedade é muito influenciada pela informática. A moeda oficial é o Bit; existem notas de B$ 50,00, B$10,00, B$5,00 e B$1,00. Você foi contratado(a) para ajudar na programação dos caixas automáticos de um grande banco das Ilhas Weblands.

1. Tarefa) Os caixas eletrônicos das Ilhas Weblands operam com todos os tipos de notas disponíveis, mantendo um estoque de cédulas para cada valor (B$ 50,00, B$10,00, B$5,00 e B$1,00). Os clientes do banco utilizam os caixas eletrônicos para efetuar retiradas de um certo número inteiro de Bits. Sua tarefa é escrever um programa que, dado o valor de Bits desejado pelo cliente, determine o número de cada uma das notas necessário para totalizar esse valor, de modo a minimizar a quantidade de cédulas entregues. Por exemplo, se o cliente deseja retirar B$50,00, basta entregar uma única nota de cinquenta Bits. Se o cliente deseja retirar B$72,00, é necessário entregar uma nota de B$50,00, duas de B$10,00 e duas de B$1,00.

2. Entrada) A entrada é composta de vários conjuntos de teste. Cada conjunto de teste é composto por uma única linha, que contém um número inteiro positivo V, que indica o valor solicitado pelo cliente. O final da entrada é indicado por V = 0.

Exemplo de Entrada

1

72

0

3. Saída) Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde n é numerado a partir de 1. Na segunda linha devem aparecer quatro inteiros I, J, K e L que representam o resultado encontrado pelo seu programa: I indica o número de cédulas de B$50,00, J indica o número de cédulas de B$10,00, K indica o número de cédulas de B$5,00 e L indica o número de cédulas de B$1,00. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

Teste 1

0 0 0 1

Teste 2

1 2 0 2

(esta saída corresponde ao exemplo de entrada acima)