

(**) 1. O número 3025 possui a seguinte característica: $30 + 25 = 55$; $55 * 55 = 3025$. Escreva um programa para obter todos os números de 4 algarismos com a mesma característica do número 3025.

(**) 2. Em 1937, o matemático Lothar Collatz propôs a seguinte conjectura, ainda não provada: dado um número natural arbitrário a_0 , a sequência gerada por:

$$a_i = a_{i-1}/2, \text{ se } a_{i-1} \text{ for par ou}$$
$$a_i = 3 \cdot a_{i-1} + 1, \text{ se } a_{i-1} \text{ for ímpar,}$$

em que a_{i-1} e a_i representam o último elemento produzido para a sequência e o próximo valor a ser gerado, respectivamente, sempre chegará ao valor 1, independente do valor inicial escolhido para a_0 .

Por exemplo, a sequência de Collatz para $a_0 = 7$ é: 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2 e 1.

Escreva um programa que leia o primeiro elemento de uma sequência de Collatz e imprima na tela a sequência a partir do elemento dado, até que se alcance o valor 1. O programa deve também mostrar o número de elementos da sequência gerada. Se o programa receber o valor 7, por exemplo, deve mostrar os 17 elementos listados acima, e dizer que gerou 17 elementos.

(**) 3. A tabela abaixo foi copiada do website da Receita Federal, e traz as alíquotas do imposto de renda de pessoa física retido na fonte para o exercício de 2013:

Base de cálculo mensal em R\$	Alíquota %	Parcela a deduzir do imposto em R\$
Até 1.637,11	-	-
De 1.637,12 até 2.453,50	7,5	122,78
De 2.453,51 até 3.271,38	15,0	306,80
De 3.271,39 até 4.087,65	22,5	552,15
Acima de 4.087,65	27,5	756,53

A base de cálculo é dada pelo salário mensal, com certas deduções, como dependentes e contribuição previdenciária. Abstraindo estes e outros detalhes, o imposto devido é calculado tomando a base de cálculo mensal, verificando a faixa na qual ela se encontra, aplicando a alíquota correspondente, e reduzindo o valor final da parcela a deduzir. Por exemplo, se a base de cálculo é de R\$10.000,00, a alíquota é de 27,5%, ou seja, R\$2750,00. Deduzimos R\$756,53 da parcela e obtemos como resultado final R\$1993,47 de imposto devido.

Escreva um programa que receba como entrada a base de cálculo mensal de um trabalhador e retorne o imposto de renda devido.

(***) 4. Escreva um programa no qual o usuário digita um número arbitrário de números inteiros positivos. Quando o usuário digitar um número menor ou igual a 0, o programa deve indicar o tamanho da maior sequência crescente observada. Por exemplo, se os números digitados forem 5, 10, 3, 2, 4, 7, 9, 8, 5, a maior sequência crescente é 2, 4, 7, 9, então o programa deve mostrar na tela que a maior sequência crescente tem 4 números. Já a sequência 10, 8, 7, 5, 2 está em ordem decrescente, portanto a maior sequência crescente observada tem tamanho 1.

(***) 5. Para evitar erros de digitação em números de grande importância, como código de uma conta bancária, geralmente se adiciona ao número um dígito verificador. Por exemplo, o número 1841 é utilizado normalmente como 18414, onde o 4 é o dígito verificador. Uma forma de se calcular um dígito verificador é:

a) Cada algarismo do número é multiplicado por um peso começando de 2 da direita para a esquerda. Para cada algarismo o peso é acrescido de 1. Somam-se os produtos obtidos.

Exemplo: $1*5 + 8*4 + 4*3 + 1*2 = 51$

b) Calcula-se o resto da divisão desta soma por 11:

Exemplo: $51 \% 11 = 7$

c) Subtrai-se de 11 o resto obtido:

Exemplo: $11 - 7 = 4$

d) Se o valor obtido for 10 ou 11, o dígito verificador será o 0, nos outros casos, o dígito verificador é o próprio valor encontrado.

Escreva um programa que leia valores inteiros entre 1 e 999 e, para cada número, imprima o seu dígito verificador. O programa é encerrado ao ser fornecido um número fora da faixa estabelecida. Para obter o dígito verificador, escreva uma função `digitoVerificador`, que recebe como parâmetro um número inteiro e retorna o seu dígito verificador.

(***) 6. Ainda considerando o método para calcular o dígito verificador da questão anterior, escreva um programa que leia um número indeterminado de valores inteiros de 10 a 9999 onde o último algarismo representa o seu dígito verificador, e imprima para cada número uma mensagem indicando se ele está correto ou não. O programa é encerrado ao ser fornecido um número fora da faixa estabelecida (10 a 9999). Para realizar o teste, escreva uma função `testaDigitoVerificador`, que recebe como parâmetro um número inteiro e retorna 1 se o número tem o dígito verificador correto e 0 do contrário. Escreva também funções auxiliares `separaDigitoVerificador` e `separaNumeroOriginal`, que separam o número original de 3 dígitos do dígito verificador, e reutilize a função `digitoVerificador` da questão anterior.

(**) 7. Escreva um programa que sorteie um número aleatório entre 0 e 500 e pergunte ao usuário qual é o "número secreto". O programa deve indicar a cada tentativa do usuário se ela é maior ou menor que o número sorteado. Quando o jogador conseguir descobrir o número secreto, o programa deve dizer quantas tentativas ele precisou fazer.

Dica: acesse <http://www.vivaolinux.com.br/dica/Gerando-numeros-aleatorios-em-C>

(**) 8. Escreva um programa que imprima losangos formados por caracteres ASCII. O usuário deve especificar o caractere que será usado, e fornecer um número n . A largura do losango no centro deve ser igual a $2n+1$ caracteres. A cada linha acima ou abaixo do centro, a largura é reduzida em 2 caracteres. Para manter o formato do losango, use espaços antes dos caracteres visíveis. Por exemplo, se o caractere é '#' e n é igual a 2, a saída do programa é:

```
  #
 ##
####
 ##
  #
```

Se o caractere é '@' e n é igual a 5, a saída do programa é:

```
  @
 @@@
 @@@@
 @@@@@
 @@@@@@
 @@@@@@
 @@@@@@
 @@@@@@
 @@@@@@
 @@@@@@
 @@@@@
 @@@@@
 @@@@
 @@@
 @
```

A chave para descobrir como resolver este exercício é se desligar da linguagem C num primeiro momento, e procurar um padrão em como os losangos são gerados. A solução para este problema precisa começar no papel. A ideia por trás do algoritmo é bem mais sofisticada que o código que a implementa.