1. Considere o polígono irregular abaixo, formado por três retângulos e um triângulo.



1. Crie a função areaRetangulo: recebe os lados de um retângulo e retorna a área
2. Crie a função hipotenusa(cateto1,cateto2) que recebe dois catetos de um triângulo retângulo e retorna a hipotenusa deste triângulo
3. Crie a função areaTotal(a, b, c, d, e) que dados os valores dos lados **a, b, c, d, e**, calcule a área total do polígono. Utilize, obrigatoriamente, as funções do item a e b, definidas anteriormente para calcular as áreas dos retângulos e do triângulo. A função também exibe a área dos retângulos e do triangulo. Lembre-se que a área de um triângulo retângulo de catetos cateto1 e cateto2 é dada pela metade da área de um retângulo de lados cateto1 e cateto2.
4. Faça um programa que pergunte ao usuário os valores de **a, b, c, d, e.** Este programa deve exibir a área de cada figura e a área total.
5. Faça um programa, utilizando adequadamente as funções abaixo (feitas nos itens a, b e c), que leia o ano (número inteiro de 4 dígitos), descubra os valores **A**, **B**, **C**, **D**, **E**, **X** e **Y**, calcule e exiba o dia e o mês da Páscoa do ano lido, caso possa ser calculado, ou a mensagem 'Impossível de determinar', caso contrário. Os valores **A**, **B**, **C**, **D** e **E** são calculados pelas fórmulas abaixo. Os valores **X** e **Y** são calculados pelas funções dos itens a e b.

**A = ano % 19 D = (19 \* A + X) % 30**

**B = ano % 4 E = (2 \* B + 4 \* C + 6 \* D + Y) % 7**

**C = ano % 7**

Tabela para utilização nos itens a e b:

|  |  |  |
| --- | --- | --- |
| Ano | X | Y |
| de 1900 a 1999 | 24 | 5 |
| de 2000 a 2099 | 24 | 5 |
| de 2100 a 2199 | 24 | 6 |
| de 2200 a 2299 | 25 | 7 |

1. Faça a função **descobre\_x** que recebe como parâmetro o ano e retorna o valor de X de acordo com **a tabela acima.**
2. Faça a função **descobre\_y** que recebe como parâmetro o ano e retorna o valor de Y de acordo com a tabela acima.
3. Faça a função **exibe\_dia\_mes** que recebe como parâmetros os valores **A**, **D** e **E** e exibe o dia e o mês da Páscoa, calculados calculados da seguinte forma:
   * Caso (D + E) > 9 🡪 DIA = (D + E - 9) e MES = 4 (Abril);
   * Caso (D + E) ≤ 9 🡪 DIA = (D + E + 22) e MES = 3 (Março).

Há dois casos particulares que ocorrem duas vezes por século:

* Quando o domingo de Páscoa cair no mês 4 (Abril) e o dia for 26, corrige-se para uma semana antes, ou seja, vai para dia 19;
* Quando o domingo de Páscoa cair no mês 4 (Abril), o dia for 25, **D** for igual a 28 e **A** for maior que 10, então o dia é corrigido para 18.

1. Crie uma função recursiva que receba um número e retorne-o como *string* (mesma tarefa da função *str*())
2. Faça uma função que recebe duas strings e retorna a concatenação da primeira com o inverso da segunda, com exceção do primeiro caractere de cada uma.

Exemplo: dadas as entradas ’abcd’ e ’efghi’, o valor de retorno ser´a ’bcdihgf’.

1. Escreva uma função que receba uma string e retorne uma concatenação da string com 4 cópias da *string* formada pelo primeiro, segundo, último e penúltimo caracteres.

Exemplo, se a entrada for ‘abcdefg’, a saída deve ser ‘abcdefhabgfabgfabgfabgf’.

1. Uma empresa aérea está fornecendo um desconto padrão para todos os seus clientes na compra de passagens do tipo inteira. Para os clientes que possuem cartão fidelidade, a empresa oferece desconto também extra, dependendo do número de tarifas do tipo meia e inteiras.
2. Faça a função **desconto\_padrao** que recebe como parâmetros o preço da tarifa normal e a quantidade de passagens do tipo inteira e retorna o valor total do desconto padrão (não é o percentual de desconto). Cálculo do valor do desconto padrão:

* Até 2 passagens inteiras 🡪 10% de desconto no preço da tarifa normal inteira
* De 3 até 5 passagens inteiras 🡪 15% de desconto no preço da tarifa normal inteira
* Acima de 5 passagens inteiras 🡪 20% de desconto no preço da tarifa normal inteira

1. Faça a função **desconto\_fidelidade** que recebe como parâmetros a quantidade de passagens do tipo meia e a quantidade de tarifas inteiras e retorna o valor total do desconto fidelidade. Cálculo do valor do desconto fidelidade:

* R$100,00 de desconto caso o número de inteiras seja menor que o número de meias
* R$200,00 de desconto caso o número de inteiras seja o dobro do número de meias
* R$100,00 de desconto caso o número de inteiras seja maior que o número de meias
* R$100,00 de desconto caso o número de inteiras seja igual ao número de meias

1. Faça um programa, utilizando adequadamente as funções acima (feitas nos itens a e b), que leia o preço da tarifa normal inteira, o tipo do cliente (‘f’ – possui cartão fidelidade, ‘n’ – não possui cartão fidelidade), a quantidade de passagens do tipo inteira e a quantidade de passagens do tipo meia e exiba o valor total do desconto e o valor total a ser pago pelo cliente na compra das passagens.

*Observação: Sabe-se que o valor da passagem do tipo meia é 50% da tarifa normal inteira*

1. Crie uma função recursiva que receba um número e exiba-o conforme exemplo:

Nº1234 Saída 1

12

123

1234

1. Criar uma função recursiva que seja capaz de intercalar os caracteres de uma string s1 com os de uma string s2. Quando os elementos de uma das duas strings terminar, concatenar os demais caracteres da string que não finalizou.

Exemplo: s1: local s2: misterio. string resultante: : lmoicsatlerio.

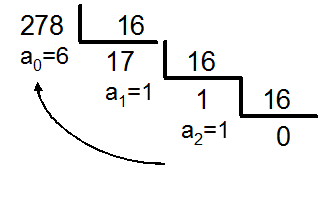
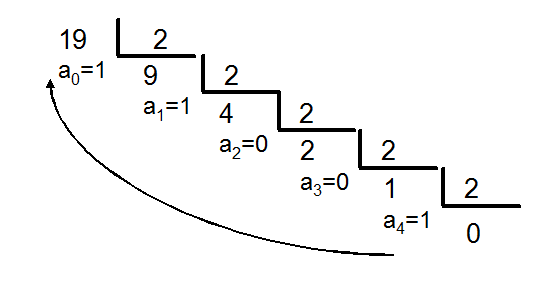
1. Observação: Para facilitar, considere que as 3 notas de cada critério serão sempre diferentes entre si. Crie uma função recursiva que receba um número n, na base 10, uma base B e retorne o número n na base B

*Dica: O número decimal deve ser dividido sucessivas vezes pela base, o resto de cada divisão ocupará sucessivamente as posições de ordem 0, 1, 2 e assim por diante, até que o resto da última divisão (que resulta em quociente 0) ocupe a posição de mais alta ordem.*

Exemplo:

19 10 == 10011 2

1. 10 == 116 16



1. Crie uma função recursiva que receba uma string e retorne outra string repetindo apenas os algarismos. Ex: prog1 => prog11
2. Crie uma função recursiva que receba duas strings e retorne True se são iguais ou False caso contrário
3. Crie uma função recursiva que receba duas palavras e retorne True caso a primeira palavra seja prefixo da segunda ou False caso contrário

Exemplo: 'Ana' e 'Anarquia' --> True

'Ana' e 'Anão' --> False

1. O número da sorte de uma pessoa irá determinar o bônus recebido. O bônus é o número da sorte \* 1000,00 reais.

a) Faça a função **acrescimo** que recebe como parâmetro o nome, um número inteiro e o sexo ('f' – feminino, 'm' - masculino) e retorna o número recebido com o acréscimo. O acréscimo é calculado da seguinte forma:

* Caso o sexo seja feminino 🡪 número\_com\_acréscimo = número recebido + 4
* Caso o sexo seja masculino 🡪 número\_com\_acréscimo = número recebido + 2

Independente do sexo, caso o nome da pessoa tenha a sequência 'sol' ou 'Sol', deverá ser somado 7 ao número\_com\_acréscimo.

b) Faça a função **numero\_sorte** que recebe como parâmetro o dia (inteiro), o mês (inteiro), o ano (inteiro de 2 dígitos) e o nome de uma pessoa (string). Esta função deverá utilizar, obrigatoriamente, a função **acrescimo** acima para calcular o dia\_com\_acréscimo e o mês\_com\_acrescimo. O número da sorte é o dígito menos significativo do resultado da soma: dia\_com\_acrescimo + mês\_com\_acrescimo + ano.

**Exemplos**:

Se dia = 30, mês = 10, ano = 80, sexo = 'f', nome = 'Marisol', teremos: 41 (30+4+7) + 21(10+4+7) + 80 = 14**2** 🡪 nº da sorte = **2**

Se dia = 30, mês = 10, ano = 80 e sexo = 'm', nome = 'Solano' teremos: 39 (30+2+7)+ 19(10+2+7) + 80 = 13**8** 🡪 nº da sorte = **8**

Se dia = 30, mês = 10, ano = 80, sexo = 'f', nome = 'Maria', teremos: 34 (30+4) + 14(10+4) + 80 = 12**8** 🡪 nº da sorte = **8**

Se dia = 30, mês = 10, ano = 80 e sexo = 'm', nome = 'Carlos' teremos: 32 (30+2)+ 12(10+2) + 80 = 12**4** 🡪 nº da sorte = **4**

c) Faça um programa, utilizando adequadamente pelo menos as funções acima (itens a e b), que pergunte ao usuário seu nome (string), o sexo ('f'-feminino, 'm' - masculino), o estado civil (‘c’ – casada, ‘o’ – outros) e a data de nascimento ('dd/mm/aa'). Caso a pessoa seja casada, o programa deverá ler também sua data de casamento ('dd/mm/aa'). O programa deverá exibir seu nome e o bônus que ela receberá de Natal. A data usada no cálculo do número da sorte depende do estado civil da pessoa:

* Para a pessoa que não é casada, o número da sorte é calculado utilizando-se o dia, o mês e o ano de nascimento.
* Para a pessoa casada, o número da sorte é o maior entre os dois valores abaixo:
* Utilizando-se o dia, o mês e o ano de nascimento
* Utilizando-se o dia, o mês e o ano de casamento

14. Uma loja oferece um bônus para seus clientes no ato de uma compra. O valor desse bônus depende do valor da última compra (uc) do cliente e de quando esta compra foi realizada. A regra para o cálculo do bônus é a seguinte:

• Se uc realizada há no máximo 3 meses, o valor do bônus será igual a 20% do valor da uc;

• Se uc realizada entre 4 e 6 meses atrás, o valor do bônus será igual a 10% do valor da uc;

• Se uc realizada há mais de 6 meses, o valor do bônus será igual a 5% do valor da uc.

*Observação*: Caso o valor do bônus seja maior do que o valor da compra atual, o valor a ser pago será igual a zero.

Escreva um programa que leia o valor da compra atual, o valor da última compra e há quantos meses essa compra foi realizada, e exiba o valor que será efetivamente pago pelo cliente, com duas casas decimais. O seu programa deverá usar, obrigatoriamente, as funções **bonusCliente** e **calculaValorComBonus**, que serão implementadas por você.

A função **bonusCliente** receberá como parâmetros o valor da última compra e há quantos meses ela foi realizada, e retornará o valor do bônus.

A função **calculaValorComBonus** receberá como parâmetros o valor da última compra, há quantos meses ela foi realizada e o valor da compra atual, e retornará o valor a ser efetivamente pago pelo cliente, que será igual ao valor da compra atual menos o valor do bônus. A função **calculaValorComBonus** terá de chamar a função **bonusCliente** para realizar o cálculo do valor a ser pago.

15. O Comitê Olímpico Internacional pretende premiar os países de acordo com o número de medalhas que conquistou. Seguindo as tabelas, a seguir, faça um programa em C que leia a quantidade de medalhas conquistadas (ouro, prata e bronze) por um país, calcule e exiba o valor do prêmio ganho.

O cálculo da pontuação deve ser realizado por uma função que recebe a quantidade de medalhas e retorna a pontuação de acordo com a tabela:

|  |  |
| --- | --- |
| Medalha | Pontuação |
| Ouro | 6 |
| Prata | 3 |
| Bronze | 1 |

O cálculo do premio também deve ser calculado por uma função que recebe a pontuação e a quantidade de medalhas de ouro, retornando o valor do prêmio de acordo com a tabela:

|  |  |
| --- | --- |
| Pontuação | Premiação |
| Até 10 pontos | R$ 50.000,00 |
| Entre 11 e 20 pontos | R$ 200.000,00 |
| Acima de 20 pontos | R$ 400.000,00 + R$ 10.000 por medalha de ouro |

16.A Biju aceita encomendas dos produtos de seu catálogo. O preço dos produtos sob encomenda depende de seu código de acordo com a seguinte regra:

os produtos de código par devem ter seu código multiplicado por R$ 0.15 e

os produtos de código ímpar devem ser multiplicados por R$ 0.04.

Exemplos:

o produto de código 2024 custa R$303,60 (2024\*0.15)

o produto de código 1033 custa R$ 41,32 (1033\*0.04).

*Obs: Um número par tem o resto da sua divisão por 2 == 0*

No entanto, a loja fornece os seguintes descontos:

1. Pelo valor total da compra:

* 25% de desconto no valor total a pagar, para encomendas que superam 1000 reais,
* 35% para as que são acima de 3000 reais
* 40% para as que são acima de 5000 reais.

1. Pelo mês da compra: (sobre o valor a pagar pela encomenda já com desconto descrito acima)

* Mês com ‘r’ : R$ 20,00 a cada R$ 500,00 em compras
* Mês sem ‘r’ e com ‘a’ : R$ 80,00 a cada R$ 500,00 em compras
* Demais meses: R$ 35,00 a cada R$500,00 em compras

*Obs*: só parcelas inteiras recebem este desconto. Por exemplo, R$ 600,00 recebe r$20,00.

Construa um programa que processe a encomenda de um cliente. É perguntado ao usuário, o código do produto, a quantidade desejada e o mês da compra. O programa deve calcular e mostrar, o valor total da encomenda sem desconto, o desconto pelo valor total da encomenda, o desconto pelo mês da compra e o valor final a pagar. Utilize as funções abaixo:

1. Faça uma função que receba o código do produto e retorne o preço unitário do produto.
2. Faça uma função que receba o valor total da encomenda e retorne o desconto pelo valor total
3. Faça uma função que receba o mês da compra e o valor a pagar (já com o desconto sobre o total) e retorne o desconto pelo mês da compra
4. Faça uma função que receba os dados da compra, o preço a pagar e o desconto e exiba na tela as saídas do programa
5. Os dois dígitos verificadores de um número inteiro com 6 algarismos (ABCDEF) formam um número inteiro com dois algarismos (XY). Eles são calculados do seguinte modo:
6. Partes Bases:
   * Parte Base 1 (PB1) = D\*8 + E\*4 + F\*3
   * Parte Base 2 (PB2) = A\*8 + B\*4 + C\*3
7. Unidade dos dígitos verificadores (Y):

* Resto da divisão de PB1 por 8, se PB1 for par;
* Resto da divisão de PB1 por 7, se PB1 for ímpar;

1. Dezena dos dígitos verificadores (X):

* Resto da divisão de PB2 por 8, se PB2 for par;
* Resto da divisão de PB2 por 7, se PB2 for ímpar.

Construa uma função, chamada **digitosVerificadores**, que receba um número inteiro (**N**) com no máximo 6 algarismos e retorne um número inteiro com dois algarismos, que represente os dígitos verificadores de **N**. Considere que um inteiro com menos de 6 algarismos possui tantos zeros à esquerda quantos forem necessários para que se complete 6 algarismos. Isto é, o inteiro 1002, para fins de cálculo dos dígitos verificadores, deve ser tratado como sendo 001002.

Para obter os dígitos verificadores de um número, obrigatoriamente, você terá de construir e usar as seguintes funções auxiliares:

* **ehMultiplo**: recebe dois números inteiros e retorna **True**, se o 1º é múltiplo do 2º, ou **False**, caso contrário;
* **algarismoDigitoVerificador**: recebe uma parte base (PB) e retorna um dos algarismos dos dígitos verificadores, de acordo com os critérios definidos nos itens 2 e 3. Esta função deve usar, obrigatoriamente, a função **ehMultiplo**;
* **parteBaseDigitoVerificador**: recebe um número com 3 algarismos e retorna a parte base desse número, de acordo com o critério definido no item 1.

Exemplos:

* Os dígitos verificadores do inteiro 123457 formam o inteiro 43.
* Os dígitos verificadores do inteiro 1002 formam o inteiro 36.