**Questão 1)** Escreva uma função **RECURSIVA**, chamada **qtAlg**, que receba uma string como parâmetro e retorne a quantidade de dígitos decimais presentes nessa string que antecedem o primeiro caractere '@' encontrado. Caso não exista nenhum caractere '@' na string, a função deve retornar -1.

Exemplos:

* qtAlg ("Dia 5/05 tem 1@ e dia 2/2 tem 2 demais ") 🡪 a função retornará 4
* qtAlg ("Dia 5/05 tem 1@ e dia 2/2 tem 2 demais, mas tem @ no texto") 🡪 a função retornará 4
* qtAlg ("Dia 5/05 não tem 1 e dia 2/2 tem 2 demais") 🡪 a função retornará -1
* qtAlg ("Dia @ 5/05 não tem 1 e dia 2/2 também") 🡪 a função retornará 0 (zero)
* Complete a função **QUESTAO1**() do template da prova, para testar sua função **qtAlg** com os valores fornecidos nos exemplos. Esta função deve exibir os resultados retornados pelas chamadas de **qtALg**.

**Questão 2)** 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:

1. Partes Bases:
   * Parte Base 1 (PB1) = D\*8 + E\*4 + F\*3
   * Parte Base 2 (PB2) = A\*8 + B\*4 + C\*3
2. 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.
* Complete a função **QUESTAO2**() do template da prova para testar sua função **digitosVerificadores** com os valores fornecidos nos exemplos. Esta função deve exibir o resultado retornado pelas chamadas da **digitosVerificadores**.

**Questão 3)**

Uma empresa gera as senhas de acesso para seus funcionários baseada nos dados por eles fornecidos:

* Nome completo (string);
* Idade (número inteiro);
* Data de aniversário (string no formato ‘dd/mm’);
* Sexo (string com um caractere: ‘m’ ou ‘f’).
* Complete a função **QUESTAO3()** fornecida no template da prova para que os dados pessoais de um funcionário sejam lidos do teclado (nome completo, idade, data do aniversário e sexo)e exiba a sua senha de acesso. Para tal, **obrigatoriamente**, esta função deverá utilizar as seguintes funções feitas por você:
* **geraCaracteres**: recebe a idade e o sexo de um funcionário e retorna uma string com dois caracteres; a string a ser retornada é definida pela seguinte regra:

Para homens (sexo = ‘m’)

) 🡪 "#\*"

) 🡪 "$%"

) 🡪 "&@"

Para mulheres (sexo = ‘f’)

) 🡪 "#%"

) 🡪 "&$"

* **geraSenhaBase**: recebe duas strings como parâmetros e retorna uma string contendo a senha base de um funcionário; a primeira string recebida é o nome do funcionário e a segunda é a data de seu aniversário (dd/mm);

A regra de formação da senha base é a seguinte:

* Gerar uma string que contenha o resultado da soma (operação sobre números inteiros) do dia com o mês de aniversário (dd+mm);
* Inserir a string no meio do nome invertido do funcionário.

Exemplo (apenas um exemplo)**:** se a função receber"Luiz Felipe" e "19/04", retornará a string "epile23F ziuL"

* **geraSenhaAcesso**: recebe o nome, a idade, a data do aniversário e o sexo de um funcionário e retorna a senha de acesso do mesmo.

A regra de formação da senha de acesso é a seguinte:

* Obtenha o 1º caractere (c1) da string retornada pela função **geraCaracteres**;
* Obtenha o 2º caractere (c2) da string retornada pela função **geraCaracteres**;
* Obtenha a senha base (sb), string retornada pela função **geraSenhaBase**;
* A senha de acesso é a concatenação das três strings (c1 + sb + c2).

Exemplo (apenas um exemplo):

Entrada de dados:

Nome: Luiz Felipe

Idade: 25

Data de aniversário: 19/04

Sexo: m

Saída gerada pelo programa:

Senha de acesso: $epile23F ziuL%