

Lista avaliativa 3

Esta é a lista avaliativa 3 da disciplina de Algoritmos e Programação Estruturada. A lista, que equivale a 3 pontos na média final, é composta de 3 questões. **Cada questão vale 1 ponto.**

Para receber o valor total de cada questão, **o código enviado deve gerar 100% das saídas corretamente, de acordo com a descrição de cada problema.** Nesse sentido, teste sua solução no marvin, lembrando-se sempre de checar a escrita correta das mensagens de saída e se foram acrescentadas as quebras de linha em cada comando *printf*. **Caso apenas uma parte das saídas para os exemplos usados para corrigir determinada solução esteja correta, o aluno receberá 0. Isso significa que somente no caso de 100% das saídas estarem de acordo com o gabarito, o aluno receberá a nota da questão, o que torna a correção binária, ou seja, ou se acerta e recebe a pontuação total da questão ou a nota para essa questão é 0.**

Soluções com erros de sintaxe (falta de “;”, nome de funções ou tipos escritos errados, etc.) ou de execução (variáveis indefinidas, escolha do identificador de tipo inadequado, etc.) também serão penalizadas e receberão nota 0. Portanto, testem o código antes de enviá-lo!

Plágio não será tolerado também! Códigos claramente iguais, com alto grau de semelhança (mesmos nomes de variáveis, lógica e estrutura praticamente idêntica, comentários iguais, etc.) receberão nota 0!

A data de entrega da lista é dia 27/11, até às 23:59. O envio deve ser feito pelo AVA e deve conter um link para uma pasta do GitHub contendo os códigos que solucionam os problemas abaixo.

Questão 1

Crie um programa que transforme um número romano e mostre sua representação em 3 bases numéricas: decimal, binária e hexadecimal.

Entrada

A entrada é composta de uma string, de até 12 caracteres, representando um número romano entre 1 e 1000. A lógica por trás da formação desses números é detalhada [aqui](#).

Saída

A saída deve ser formada por três linhas, contendo respectivamente:

- A representação binária do número da entrada;
- A representação decimal do número da entrada;
- E a representação hexadecimal do número da entrada.

As bases numéricas acima são explicadas em detalhe [aqui](#).

Observações

- Valores nulos (0s) a esquerda nas saídas devem ser desconsiderados;
- Lembre-se de acrescentar uma quebra de linha (“\n”) após imprimir a última linha.

Exemplo de entrada	Saída
M	M na base 2: 1111101000 M na base 10: 1000 M na base 16: 3e8
XLII	XLII na base 2: 101010 XLII na base 10: 42 XLII na base 16: 2a

Questão 2

Após bastante esforço e estudo, Charlinho conseguiu se formar e começou a trabalhar na área de programação. Seu emprego pagava um salário muito bom, mas ele sabia que seria importante poupar e investir parte do dinheiro no começo de cada mês para garantir uma aposentadoria mais tranquila.

Dessa forma, faça um programa que simule o montante que Charlinho terá ao fim de cada mês após investir parte da sua remuneração. Para calcular o valor ao fim de cada mês, use a seguinte fórmula:

$$V = A * (1 + i) * \left[\frac{(1 + i)^t - 1}{i} \right]$$

Onde:
V = Valor total ao fim do mês
A = Aporte mensal
i = taxa de juros mensal
t = tempo em meses investindo

Entrada

- A entrada é composta de 3 valores:
- Um valor inteiro (entre 1 e 240) que é o número de meses poupando e investindo;
 - Um valor de ponto flutuante de dupla precisão (entre 1 e 5000), que é o aporte mensal;
 - Um valor de ponto flutuante de dupla precisão (entre 0.01 e 0.2), que é a taxa de retorno mensal dos investimentos;

Saída

A saída deve mostrar o valor do montante **ao fim de cada mês de investimento**, no formato mostrado nos exemplos abaixo. **Lembre-se que todas as linhas de saída devem ter um “\n” após a mensagem.**

Exemplo de entrada	Saída
1 1000 0.03	Montante ao fim do mes 1: R\$ 1030.00
12 1000 0.03	Montante ao fim do mes 1: R\$ 1030.00 Montante ao fim do mes 2: R\$ 2090.90 Montante ao fim do mes 3: R\$ 3183.63 Montante ao fim do mes 4: R\$ 4309.14 Montante ao fim do mes 5: R\$ 5468.41 Montante ao fim do mes 6: R\$ 6662.46 Montante ao fim do mes 7: R\$ 7892.34 Montante ao fim do mes 8: R\$ 9159.11 Montante ao fim do mes 9: R\$ 10463.88 Montante ao fim do mes 10: R\$ 11807.80 Montante ao fim do mes 11: R\$ 13192.03 Montante ao fim do mes 12: R\$ 14617.79

Questão 3

Uma cidade vai passar a utilizar o rodízio de veículos em horários de pico, para minimizar o congestionamento em algumas vias. Essa estratégia estabelece uma escala que determina em quais dias um veículo com determinada placa pode transitar. Para isso, é necessária a criação de um programa que vai checar automaticamente se determinada placa pode ou não transitar naquele dia. O programa deve se basear na seguinte escala:

- Na segunda-feira, é proibida a circulação de veículos com placas terminadas em 0 e 1;
- Na terça-feira, é proibida a circulação de veículos com placas terminadas em 2 e 3;
- Na quarta-feira, é proibida a circulação de veículos com placas terminadas em 4 e 5;
- Na quinta-feira, é proibida a circulação de veículos com placas terminadas em 6 e 7;
- Na sexta-feira, é proibida a circulação de veículos com placas terminadas em 8 e 9;
- Para sábado e domingo, não há proibição.

Entrada

A entrada é composta de 2 linhas, uma com a placa consultada e outra com um dia da semana. **As placas podem estar no formato antigo (LLL-NNNN), onde L é uma letra e N é um dígito, ou no formato novo, do Mercosul (LLLNLNN). Mas também pode haver a entrada de uma placa com formato inválido, o que deve ser checado pelo programa.**

Os dias da semana devem estar com os caracteres em maiúsculo e sem acentuação, obedecendo o seguinte padrão: SEGUNDA-FEIRA, TERCA-FEIRA, QUARTA-FEIRA, QUINTA-FEIRA, SEXTA-FEIRA, SABADO e DOMINGO. **Também pode haver a entrada de um dia em um formato impróprio, o que deve ser validado.**

Saída

A saída deve mostrar se a placa de entrada pode circular ou não no dia inserido, conforme mostrado abaixo. Caso a placa digitada seja inválida, a mensagem “Placa invalida” deve ser mostrada. Se o dia da semana estiver errado ou em um formato inválido, a mensagem “Dia da semana invalido” deve aparecer na saída. **Todas as linhas de saída devem ter um “\n” após a mensagem.**

Exemplo de entrada	Saída
JJJ-4242 QUINTA-FEIRA	JJJ-4242 pode circular quinta-feira
AAA-1010 SEGUNDA-FEIRA	AAA-1010 nao pode circular segunda-feira
XX-1111 TERCA-FEIRA	Placa invalida
BRA1X42 SABADO	Nao ha proibicao no fim de semana
AAA1X00 DOMINGO	Nao ha proibicao no fim de semana
BRA1X42 sábado	Dia da semana invalido
AAAAA sábado	Placa invalida Dia da semana invalido