**Lógica de Programação**

**Lista EXTRA de atividades**

**Estruturas de repetição - enquanto-faca**

**Atividade: Desenvolvimento técnico em programação**

**Tema : Fundamento em lógica de programação**

**Indicadores associados**

**3 - Codifica programas computacionais utilizando lógica de programação e respeitando boas práticas de programação.**

**5 - Desenvolver capacidades linguísticas de modo a saber usar adequadamente a linguagem oral e escrita em diferentes situações e contextos.**

**6 - Conhecer o caráter do conhecimento científico aplicando a metodologia científica e utilizando redação acadêmica na realização da pesquisa, na escolha de métodos, técnicas e instrumentos de pesquisa.**

**8 - Utilizar estruturas de dados definindo-as e aplicando-as adequadamente nos programas.**

**ATENÇÃO:** ESTA LISTA DEVE SER FEITA APENAS COM A SOLICITAÇÃO DE UM ORIENTADOR

**OBJETOS DE SOLUÇÃO**

**Observação geral:** Em todos os programas abaixo, você deve realizar a validação dos dados preenchidos pelo usuário dentro da coerência de cada valor (não permitindo valores negativos onde não puder, não permitindo em um menu que o usuário escolha valores de opções inexistentes, e assim por diante). Essa validação deve ser feita com o Enquanto-Faça, pois primeiro você irá solicitar o valor que o usuário precisa informar e caso o usuário não informe corretamente, você deve informar o erro e solicitar o valor novamente até que o usuário não erre.

1. Desenvolva um programa que solicite o valor da fatura de cartão de crédito do usuário, os juros aplicados mensalmente pelo banco e o valor que o usuário pode pagar mensalmente. Verifique em seguida se o valor pago pelo usuário irá realmente diminuir o valor da próxima fatura (usar os 2 passos abaixo pode ajudar você a pensar sobre como fazer essa verificação), e se isso não acontecer, peça que ele informe um valor maior pois esse não fará a fatura diminuir, e teste novamente esse valor. Depois de garantir que não vai acontecer um loop infinito, faça uma estrutura de repetição que calcule o pagamento da fatura a cada mês, fazendo os seguintes passos:
   1. Acrescente os juros ao saldo da fatura restante, gerando o valor de fatura do mês atual;
   2. Subtraia o valor que o usuário informou que pagaria desse novo valor, gerando o valor de fatura restante para o próximo mês.

A repetição deve acontecer até que a fatura restante seja zero ou negativa, e você deve considerar que o usuário não fará mais gastos nesse cartão.

Por fim, exiba quantos meses foram necessários para zerar a fatura, bem como o valor da última fatura.

Exemplo:

* Valor da fatura: 3000,00
* Valor mensal a ser pago: 500,00
* Valor dos juros: 5%

Resultado:

O valor mensal a ser pago será suficiente para diminuir o valor da fatura.

Fatura mês 1: 3000.00

Fatura mês 2: 2625.00

Fatura mês 3: 2231.25

Fatura mês 4: 1817.81

Fatura mês 5: 1383.70

Fatura mês 6: 927.89

Fatura mês 7: 449.28

O valor será pago em 7 meses.

1. O prefeito da cidade de “Townsville” deseja fazer uma pesquisa entre seus habitantes. Desenvolva uma solução para coletar dados sobre a quantidade de moradores e a renda total de cada residência, sendo que a cada residência visitada o programa deve solicitar se há uma nova residência a ser visitada. Após as leituras, deve-se exibir:
   1. Média de salário da população
   2. Média do número de residentes por residência
   3. Percentual de residências com renda menor do que R$ 1500,00
2. Desenvolva um jogo chamado Portas da Esperança. O objetivo do jogo é passar pelo maior número de portas que conseguir. Cada fase do jogo possui 3 portas, sendo que duas portas levam o jogador para a próxima fase e a outra está bloqueada e faz ele perder. O usuário começa com 1 vida, e a cada 10 fases que ele passar, ganha mais uma. Seu programa deve executar cada fase informando a fase em que o usuário está e quantas vidas ele tem, depois pedir que o usuário escolha uma porta, definir por sorteio (lembre-se do randi) a porta que o fará perder e verificar se o usuário passou de fase ou não. Esse processo deve se repetir a cada fase. Caso o usuário selecione a porta bloqueada e tenha ao menos uma vida, ele perde uma vida mas vai para a próxima fase, mas se ele não tiver nenhuma vida restante ao errar, o programa deve informar quantas fases o usuário passou de uma forma criativa (pense que é um jogo).