

## Prática 1A: Conta poupança

Uma conta poupança foi aberta com um depósito de R\$789,54, com rendimentos 0.56% de juros ao mês. No início do segundo mês, R\$303,20 reais foram depositados nessa conta poupança. No início do terceiro mês, R\$58,25 reais foram *retirados* da conta. Implemente um programa que imprima o crédito em conta no início do quarto mês.

- **Entrada:** Seu programa deve inicializar dados em variáveis locais.
- **Saída:** Seu programa deve imprimir um número de ponto flutuante [float].
- **Gabarito:** 1050.91

## Prática 1B: Índice de massa corporal

Brutus e Olívia foram ao médico, que disse a eles que ambos estão fora do peso ideal. Ambos discordaram veementemente da armação do médico. Para provar que estava certo, o médico mostrou o Índice de Massa Corporal (IMC) de ambos, considerando que Brutus tem 1,84m e pesa 122kg e Olívia tem 1,76m e pesa 45kg.

Implemente um programa para mostrar o IMC de Brutus e Olívia e qual é a quantidade mínima de quilos Brutus deve perder e a que Olívia deve ganhar para atingirem um peso saudável segundo a classificação do IMC. A tabela de classificação do IMC está abaixo:

IMC	Classificação
< 16	Magreza grave
16 a 17	Magreza moderada
17 a 18.5	Magreza leve
18.5 a 25	Saudável
25 a 30	Sobrepeso
30 a 35	Obesidade grau 1
35 a 40	Obesidade grau 2 (severa)
≥ 40	Obesidade grau 3 (mórbida)

Tabela 1: Classificação de Índice de Massa Corporal

Para calcular o IMC, utilize a seguinte fórmula:

$$\text{IMC} = \text{peso} \div \text{altura}^2$$

- **Entrada:** Seu programa deve inicializar os dados em variáveis locais.

- **Saída:** Seu programa deve imprimir uma única linha com quatro números de ponto flutuante [float] separados por espaço. Os números impressos devem ser, nesta ordem: IMC do Brutus, IMC da Olivia, quantidade de quilos que o Brutus deve perder e quantidade de quilos que a Olivia deve ganhar. Cada número deve ser impresso com duas casas decimais. Dica: A chamada à função para imprimir os resultados deve seguir o seguinte esquema:

```
printf("%.2f %.2f %.2f %.2f\n", imc_brutus, imc_olivia,  
                                         perda_brutus, ganho_olivia);
```

- **Gabarito:** 36.03 14.53 37.36 12.31

## Prática 1C: Quociente e resto com somas e subtrações

Utilizando apenas somas e subtrações, implemente um programa que dados dois números imprime o quociente e o resto da divisão do primeiro número pelo segundo. Projete primeiro o algoritmo antes de iniciar a implementação.

- **Entrada:** Seu programa irá receber os dados de um arquivo de texto. Utilize o esqueleto disponibilizado no Moodle que lê a entrada e coloca o primeiro número na variável [A] e o segundo número na variável [B].
- **Saída:** Seu programa deve imprimir uma linha com dois números inteiros separados por espaço. O primeiro número deve ser o quociente da divisão de A por B e o segundo número deve ser o resto.

## Desafio: Calculador de dígito verificador de CPF

Dígito verificador ou algarismo de controle é um mecanismo de autenticação utilizado para verificar a validade e a autenticidade de um valor numérico, evitando dessa forma fraudes, erros de transmissão ou erros de digitação. Dígitos verificadores são um ou mais algarismos acrescentados ao valor original e calculados a partir deste através de um determinado algoritmo. Números de documentos de identificação, de matrícula, cartões de crédito e quaisquer outros códigos numéricos que necessitem de maior segurança utilizam dígitos verificadores. Essa prática consiste em calcular o dígito verificador do CPF. Dado um CPF como um número, sem os dois últimos dígitos verificadores, seu programa deve imprimir o CPF com os dígitos verificadores. Você pode usar o seu CPF e de parentes e colegas para testar o programa. A seguir, as regras para o cálculo do dígito verificador são definidas através de um exemplo. É utilizado como exemplo o número: 123456789.

1. Calcule a soma dos produtos dos nove dígitos utilizando peso 2 para unidade, peso 3 para dezena, peso 4 para centena e assim sucessivamente.

Exemplo:  $9*2 + 8*3 + 7*4 + 6*5 + 5*6 + 4*7 + 3*8 + 2*9 + 1*10 = 210$

2. A dezena do número verificador é 0 caso o resto da divisão por 11 da soma dos produtos acima seja 0 ou 1. Caso contrário, a dezena corresponde a subtrair de 11 o resto da divisão por 11 da soma dos produtos.

Exemplo: O resto da divisão de 210 por 11 é 1 então a dezena do número verificador é 0.

3. Calcule a soma dos produtos dos dez dígitos, onde o dígito menos significativo passa a ser a dezena do número verificador, calculada no passo 2. Utilize os seguintes pesos: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11.

Exemplo:  $2*0 + 3*9 + 4*8 + 5*7 + 6*6 + 7*5 + 8*4 + 9*3 + 10*2 + 11*1 = 255$ .

4. A unidade do número verificador é 0 caso o resto da divisão da soma dos produtos acima seja 0 ou 1. Caso contrário, a unidade corresponde a 11 menos o resto da divisão por 11 da soma dos produtos.

Exemplo: O resto da divisão de 255 por 11 é 2; a unidade do número verificador é 9.

Dica: Para resolver esse problema, você precisa aprender dois conceitos que ainda não foram vistos: calcular o resto da divisão entre dois números utilizando o operador de módulo e processamento condicional. Para o primeiro conceito, procure saber como funciona o operador de módulo [%]. Para o segundo, procure saber como funciona expressões condicionais [if, else] na linguagem C. O aprendizado de uma linguagem depende muito da sua iniciativa de procurar soluções. Esse exercício é uma forma de incentivar a auto-didática.