

Trabalho 3

LEIA ATENTAMENTE AS REGRAS E OS ENUNCIADOS

R E G R A S

- O trabalho deverá ser realizado individualmente.
- O trabalho deverá ser enviado para o Google classroom até o dia 07/05/2021 (sexta-feira).
- A data de entrega não será adiada.
- Os 3 programas solicitados (arquivos .CPP) deverão ser compactados em um único arquivo (ZIP ou RAR) com o nome e sobrenome do aluno.
- Os programas (arquivos .CPP) deverão ter os nomes conforme definido nos enunciados.
- Não serão aceitos trabalhos enviados por email.
- Trabalhos com estruturas e/ou organizações semelhantes (plágio) serão penalizados com a nota zero.
- O programa que não obedecer às restrições estabelecidas receberá zero.

E N U N C I A D O S

1) Programa: caracteres.cpp (3,0 pontos)

Ler um inteiro **n** ($n \geq 3$) e, em seguida, ler um vetor de n caracteres. Por fim, converta os caracteres minúsculos em maiúsculos, os caracteres maiúsculos em minúsculos, e imprima o vetor resultante e a quantidade de conversões realizadas. O programa deverá ter, obrigatoriamente, 3 funções:

- a) **le_caracteres**: para ler o vetor de caracteres e retornar o vetor preenchido.
- b) **converte_caracteres**: que recebe o vetor lido na função anterior e retorna o vetor convertido e a quantidade de conversões.
- c) **imprime_caracteres**: que recebe o vetor de caracteres e imprime seu conteúdo em uma única linha.

Restrições:

- a) A função main deverá ler o valor n, chamar as funções definidas acima e imprimir a quantidade de conversões.
- b) O programa poderá ter mais funções do que o especificado no enunciado, a seu critério.
- c) Não poderão ser usadas variáveis globais.

2) Programa: desvio_padrao.cpp (3,0 pontos)

Ler um valor inteiro **n** ($n > 2$) e outro valor inteiro **k** ($2 \leq k \leq n$ e **k** divisor de **n**) e leia um vetor **v1** de inteiros com **n** elementos. Em seguida, crie um segundo vetor **v2** com o desvio padrão dos valores de **v1** agrupando esses valores de **k** em **k** elementos. Por fim, imprima o conteúdo de **v2**. O programa deverá ter, obrigatoriamente, as seguintes funções:

- a) **le_vetor**: para ler o vetor de inteiros.
- b) **imprime_vetor**: para imprimir o vetor de inteiros.
- c) **calcula_dp**: para percorrer o vetor **v1** de **k** em **k** elementos e preencher o vetor **v2** com os desvios-padrão.
- d) **desvio_padrao**: para calcular o desvio padrão de um grupo de **k** elementos.

Dica: a função definida na letra **c** deverá chamar a função definida na letra **d**.

Exemplo 1:

```
n = 12
k = 3
v1 = [12, 3, 45, 9, 28, 4, 8, 58, 19, 72, 84, 36]
v2 = [ 22.11,      16.66,      26.27,      24.98]
```

Exemplo 2:

```
n = 12
k = 6
v1 = [12, 3, 45, 9, 28, 4, 8, 58, 19, 72, 84, 36]
v2 = [      16.48,      30.12]
```

Exemplo 3:

```
n = 12
k = 2
v1 = [12, 3, 45, 9, 28, 4, 8, 58, 19, 72, 84, 36]
v2 = [ 6.36, 25.46, 16.97, 25.35, 37.48, 33.94]
```

Restrições:

- a) $n > 2$.
- b) $2 \leq k \leq n$ e **k** divisor de **n**.
- c) O vetor **v2** deverá ter o tamanho exato para armazenar os desvios-padrão.
- d) A função **main** deverá ler os valores de **k** e **n** e chamar as funções.
- e) Não poderão ser usadas variáveis globais.
- f) O programa poderá ter mais funções do que o especificado no enunciado, a seu critério.

$$\text{desvio padrão} = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{média})^2}{n - 1}}$$

3) Programa: financiamento.cpp (4,0)

Uma financiadora oferece empréstimos para pessoas físicas. O cálculo dos juros mensais é feito de acordo com a seguinte fórmula:

$$juros\ mensais = 1,8 + (fator\ de\ risco \times 0,2) \quad (\text{Fórmula I})$$

onde o fator de risco do cliente é calculado de acordo com as tabelas 1 e 2:

Tabela 1: Fator de Risco (1ª parte)

Idade	Fator
De 18 a 30 anos	3
De 31 a 50 anos	6
De 51 a 60 anos	9
Acima de 60 anos	12

Tabela 2: Fator de Risco (2ª parte)

Patrimônio	Fator
Até R\$ 50.000,00	fator = fator + 2
De R\$ 50.000,01 a R\$ 200.000,00	fator = fator + 1
De R\$ 200.000,01 a R\$ 1.000.000,00	fator = fator – 1
Acima de R\$ 1.000.000,00	fator = fator – 2

De posse dos juros mensais (fórmula I), o valor da mensalidade é calculado de acordo com a fórmula dos juros compostos:

$$mensalidade = \frac{(valor\ do\ empréstimo \times (1 + juros\ mensais)^{anos})}{número\ de\ parcelas} \quad (\text{Fórmula II})$$

Exemplo: para um empréstimo de R\$ 5.000,00 a serem pagos em 18 meses por um cliente com fator de risco 3, teremos:

$$juros\ mensais = 1,8 + (3 \times 0,2) = 2,4$$

$$mensalidade = \frac{(5000 \times (1 + 0,024)^{1,5})}{18} = 287,837$$

Ou seja, o cliente vai pagar 18 mensalidades de R\$ 287,84 com juros de 2,4% ao mês. É importante notar que a quantidade de meses (18) foi convertida para anos (1,5) para usar a fórmula.

Leia o dia, mês e ano de nascimento do cliente, o valor do patrimônio, o valor do empréstimo e a quantidade de meses de pagamento e imprima o fator de risco, o valor dos juros ao mês e a mensalidade. O programa deverá ter, obrigatoriamente, as seguintes funções:

- a) **le_dados**: para ler os dados de entrada. Deverá ser uma única função que retorna todos os dados informados. Os dados deverão ser validados pela função de acordo com as restrições. Por exemplo: se o usuário digitar o valor do empréstimo inválido, a função deverá informar o erro e solicitar novamente.
- b) **calcula_idade**: para calcular e retornar a idade do cliente a partir da data de nascimento.
- c) **fator_de_risco**: para calcular e retornar o fator de risco do cliente de acordo com as tabelas 1 e 2.
- d) **calcula_juros_mensalidade**: para calcular e retornar os juros ao mês e a mensalidade, de acordo com as fórmulas I e II.

Restrições:

- a) Dia, mês e ano devem formar uma data de nascimento válida.
- b) Valor de patrimônio ≥ 0 .
- c) O valor mínimo do empréstimo é de R\$ 1.000,00.
- d) O número mínimo de parcelas do empréstimo é 2.
- e) Só podem tomar empréstimos clientes com 18 anos ou mais.
- f) A função main deverá chamar as funções definidas acima e imprimir os dados conforme o enunciado.
- g) Não poderão ser usadas variáveis globais.
- h) O programa poderá ter mais funções do que o especificado no enunciado, a seu critério.

Dica 1: para verificar se dia, mês e ano de nascimento constituem uma data válida, use as funções implementadas nos exercícios 8.14, 8.15 e 8.16 da parte 8 do material de aula.

Dica 2: para recuperar a data atual do computador e calcular a idade do cliente, use a função a seguir:

```
#include <time.h>

void data_atual(int &dia, int &mes, int &ano) {
    time_t t = time(NULL);
    struct tm lt = *localtime(&t);

    ano = lt.tm_year + 1900;
    mes = lt.tm_mon + 1;
    dia = lt.tm_mday;
}
```