

Curso: **Análise e Desenvolvimento de Sistemas - 2018**

Disciplina: **Linguagem de Programação**

**Lista de exercícios 3**

**Laços de repetição**

Desenvolva os algoritmos a seguir utilizando a Linguagem C:

- 1) (BACKES, 2012) Faça um programa que determine o mostre os cinco primeiros múltiplos de 3, considerando números maiores que 0. Salve o projeto com o nome **"01\_multiplos\_de\_3"**.
- 2) Escreva um algoritmo para imprimir os números de 1 (inclusive) a 10 (inclusive) em ordem crescente. Salve o projeto com o nome **"02\_imprimir\_numeros\_de\_1\_a\_10\_crescente"**.
- 3) Escreva um algoritmo para imprimir os números de 1 (inclusive) a 10 (inclusive) em ordem decrescente. Salve o projeto com o nome **"03\_imprimir\_numeros\_de\_1\_a\_10\_decrescente"**.
- 4) (BACKES, 2012) Escreva um programa que escreva na tela, de 1 até 100, de 1 em 1, 3 vezes. A primeira vez deve usar a estrutura de repetição for, a segunda while, e a terceira do while. Salve o projeto com o nome **"04\_imprimir\_de\_1\_a\_100"**.
- 5) (BACKES, 2012) Faça um algoritmo utilizando o comando while que mostra uma contagem regressiva na tela, iniciando em 10 e terminando em 0. Mostrar uma mensagem "FIM!" após a contagem. Salve o projeto com o nome **"05\_contagem\_regressiva"**.
- 6) (BACKES, 2012) Escreva um programa que declare um inteiro, inicialize-o com 0, e incremente-o de 1000 em 1000, imprimindo seu valor na tela, até que seu valor seja 100000 (cem mil). Salve o projeto com o nome **"06\_incremento"**.
- 7) (BACKES, 2012) Faça um programa que peça ao usuário para digitar 10 valores e some-os. Salve o projeto com o nome **"07\_soma\_de\_10\_valores"**.
- 8) (BACKES, 2012) Faça um programa que leia 10 inteiros e imprima sua média. Salve o projeto com o nome **"08\_media\_de\_10\_inteiros"**.
- 9) (BACKES, 2012) Faça um programa que leia 10 inteiros positivos, ignorando não positivos, e imprima sua média. Salve o projeto com o nome **"09\_media\_de\_10\_inteiros\_positivos"**.
- 10) (BACKES, 2012) Escreva um programa que leia 10 números e escreva o menor valor lido e o maior valor lido. Salve o projeto com o nome **"10\_menor\_e\_maior\_lidos"**.

11) (BACKES, 2012) Faça um programa que leia um número inteiro N e depois imprima os N primeiros números naturais ímpares. Salve o projeto com o nome “11\_imprimir\_impares”.

12) (BACKES, 2012) Faça um programa que calcule e mostre a soma dos 50 primeiros números pares. Salve o projeto com o nome “12\_soma\_50\_numeros\_impares”.

13) Escreva um programa que apresente quatro opções: (a) consulta saldo, (b) saque e (c) depósito e (d) sair. O saldo deve iniciar em R\$ 0,00. A cada saque ou depósito o valor do saldo deve ser atualizado. Salve o projeto com o nome “13\_banco”.

14) Escreva um algoritmo para imprimir os números de 1 (inclusive) a 10 (inclusive) em ordem crescente e em ordem decrescente, conforme informado pelo usuário. Salve o projeto com o nome “09\_imprimir\_numeros\_de\_1\_a\_10\_crescente\_decrescente”.

15) (BACKES, 2012) Faça um programa que leia um número inteiro positivo N e imprima todos os números naturais de 0 até N em ordem crescente. Salve o projeto com o nome “15\_imprimir\_de\_0\_a\_n\_crescente”.

16) (BACKES, 2012) Faça um programa que leia um número inteiro positivo N e imprima todos os números naturais de 0 até N em ordem decrescente. Salve o projeto com o nome “16\_imprimir\_de\_0\_a\_n\_decrescente”.

17) (BACKES, 2012) Faça um programa que leia um número inteiro positivo par N e imprima todos os números pares de 0 até N em ordem crescente. Salve o projeto com o nome “17\_imprimir\_pares\_de\_0\_a\_n\_crescente”.

18) (BACKES, 2012) Faça um programa que leia um número inteiro positivo par N e imprima todos os números pares de 0 até N em ordem decrescente. Salve o projeto com o nome “18\_imprimir\_pares\_de\_0\_a\_decrescente”.

19) (BACKES, 2012) Faça um programa que leia um número inteiro positivo ímpar N e imprima todos os números ímpares de 1 até N em ordem crescente. Salve o projeto com o nome “19\_imprimir\_impares\_de\_1\_a\_n\_crescente”.

20) (BACKES, 2012) Faça um programa que leia um número inteiro positivo ímpar N e imprima todos os números ímpares de 1 até N em ordem decrescente. Salve o projeto com o nome “20\_imprimir\_impares\_de\_1\_a\_decrescente”.

21) (BACKES, 2012) Faça um programa que leia um número inteiro positivo n e calcule a soma dos n primeiros números naturais. Salve o projeto com o nome “21\_soma\_dos\_n\_numeros”.

22) (CARVALHO, 2007) Escreva um algoritmo para ler 2 valores e se o segundo valor informado for zero, deve ser lido um novo valor, ou seja, para o segundo valor não pode ser

aceito o valor zero e imprimir o resultado da divisão do primeiro valor lido pelo segundo valor lido. Salve o projeto com o nome “22\_ler\_2\_valores\_e\_dividir”.

23) (BACKES, 2012) Escreva um algoritmo que leia certa quantidade de números e imprima o maior deles e quantas vezes o maior número foi lido. A quantidade de números a serem lidos deve ser fornecida pelo usuário. Salve o projeto com o nome “23\_imprimir\_maior\_numero”.

24) (BACKES, 2012) Escreva um algoritmo que leia um número inteiro entre 100 e 999 e imprima na saída cada um dos algarismos que compõem o número. Salve o projeto com o nome “24\_imprimir\_algarismos”.

25) (BACKES, 2012) Faça um programa que receba dois números. Calcule e mostre:

- A soma dos números pares desse intervalo de números, incluindo os números digitados;
- A multiplicação dos números ímpares desse intervalo, incluindo os digitados;

Salve o projeto com o nome “25\_soma\_e\_multiplicacao”.

26) (BACKES, 2012) Escreva um programa completo que permita a qualquer aluno introduzir, pelo teclado, uma sequência arbitrária de notas (válidas no intervalo de 10 a 20) e que mostre na tela, como resultado, a correspondente média aritmética. O número de notas com que o aluno pretenda efetuar o cálculo não será fornecido ao programa, o qual terminará quando for introduzido um valor que não seja válido como nota de aprovação. Salve o projeto com o nome “26\_leitura\_de\_notas”.

27) (CARVALHO, 2007) Ler um valor N e imprimir todos os valores inteiros entre 1 (inclusive) e N (inclusive). Considere que o N será sempre maior que ZERO. Salve o projeto com o nome “27\_imprimir\_de\_1\_a\_n”.

28) (PUGA & RISSETI, 2016) Escreva um algoritmo que calcule e exiba a tabuada, até um determinado número "n", fornecido pelo usuário, lembrando que, se o número fornecido é 4, deve ser gerada a tabuada do 1, 2, 3 e 4, com as operações de multiplicação e o resultado no formato:

1 x 1 = 1

1 x 2 = 2

1 x 3 = 3

...

Salve o projeto com o nome “28\_tabuada”.

29) (BACKES, 2012) Faça um algoritmo que leia um número positivo e imprima seus divisores. Salve o projeto com o nome “29\_divisores”.

30) (BACKES, 2012) Escreva um programa que leia um número inteiro e calcule a soma de todos os divisores desse número, com exceção dele próprio. Ex: a soma dos divisores do

número 66 é  $1 + 2 + 3 + 6 + 11 + 22 + 33 = 78$ . Salve o projeto com o nome “30\_soma\_dos\_divisores”.

31) (BACKES, 2012) Faça um programa que some todos os números naturais abaixo de 1000 que são múltiplos de 3 ou 5. Salve o projeto com o nome “31\_soma\_multiplos\_de\_3\_ou\_5”.

32) (BACKES, 2012) Faça um algoritmo que encontre o primeiro múltiplo de 11, 13 ou 17 após um número dado. Salve o projeto com o nome “32\_encontrar\_multiplos”.

33) (BACKES, 2012) Em Matemática, o número harmônico designado por  $H(n)$  define-se como sendo a soma da série harmônica:

$$H(n) = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$$

Faça um programa que leia um valor  $n$  inteiro e positivo e apresente o valor de  $H(n)$ . Salve o projeto com o nome “33\_serie\_harmonica”.

34) (BACKES, 2012) Faça um programa que leia um valor  $N$  inteiro e positivo, calcule e mostre o valor  $E$ , conforme a fórmula a seguir:

$$E = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/N!$$

Salve o projeto com o nome “34\_serie\_harmonica\_fatorial”.

35) (BACKES, 2012) Faça programas para calcular as seguintes sequências:

$$1 + 2 + 3 + 4 + 5 + \dots + n$$

$$1 - 2 + 3 - 4 + 5 + \dots + (2n - 1)$$

$$1 + 3 + 5 + 7 + \dots + (2n - 1)$$

Salve o projeto com o nome “25\_sequencias”.

36) (PUGA & RISSETI, 2016) Pedro tem 1,50 m e cresce 2 cm por ano, enquanto Lucas tem 1,10 m e cresce 3 cm por ano. Construa um algoritmo que calcule e imprima quantos anos serão necessários para que:

a) Lucas e Pedro tenham o mesmo tamanho.

b) Lucas seja maior que Pedro.

Salve o projeto com o nome “36\_crescimento”.

37) (PUGA & RISSETI, 2016) A partir de um conjunto de números inteiros sequenciais, obtidos com base em dados fornecidos pelo usuário (número inicial e final), identifique e apresente:

a) A quantidade de números inteiros e positivos.

b) A quantidade de números pares.

c) A quantidade de números ímpares.

d) A quantidade de números ímpares e divisíveis por 3 e 7.

e) A respectiva média para cada um dos itens.

Salve o projeto com o nome “27\_classificacao\_numeros”.

38) (PUGA & RISSETI, 2016) Considerando a sequência de Fibonacci (1, 1, 2, 3, 5, 8, 13, ...,  $n$ ), escreva um algoritmo para gerar esta sequência até o  $n$ ésimo termo, fornecido pelo usuário. Por exemplo, se o usuário digitou o número 40, deverão ser gerados os primeiros 40 números. Salve o projeto com o nome “38\_fibonnaci”.

39) (PUGA & RISSETI, 2016) Escreva um algoritmo que, a partir de um número  $n$  fornecido pelo usuário, execute os cálculos a seguir, enquanto este  $n$  for diferente de 1:

a) Se “ $n$ ” for par,  $n = n / 2$ .

b) Se “ $n$ ” for ímpar,  $n = n * 3 + 1$

Salve o projeto com o nome “39\_calculos”.

40) (BACKES, 2012) Faça um programa que simula o lançamento de dois dados,  $d1$  e  $d2$ ,  $n$  vezes, e tem como saída o número de cada dado e a relação entre eles ( $>$ ,  $<$ ,  $=$ ) de cada lançamento. Salve o projeto com o nome “40\_dados”.

41) (BACKES, 2012) Dados  $n$  e dois números inteiros positivos,  $i$  e  $j$ , diferentes de 0, imprimir em ordem crescente os  $n$  primeiros naturais que são múltiplos de  $i$  ou de  $j$  e ou de ambos. Exemplo: Para  $n = 6$ ,  $i = 2$  e  $j = 3$  a saída deverá ser: 0,2,3,4,6,8. Salve o projeto com o nome “41\_multiplos\_de\_i\_e\_j”.

42) (BACKES, 2012) Faça um programa que some os números ímpares contidos em um intervalo definido pelo usuário. O usuário define o valor inicial do intervalo e o valor final deste intervalo e o programa deve somar todos os números ímpares contidos neste intervalo. Caso o usuário digite um intervalo inválido (começando por um valor maior que o valor final) deve ser escrito uma mensagem de erro na tela, “Intervalo de valores inválido” e o programa termina. Exemplo de tela de saída:

Digite o valor inicial: 5

Digite o valor final: 10

Soma dos ímpares neste intervalo: 21

Salve o projeto com o nome “42\_soma\_dos\_impares\_num\_intervalo”.

43) (BACKES, 2012) Escreva um programa que verifique quais números entre 1000 e 9999 (inclusive) possuem a propriedade seguinte: a soma dos dois dígitos de mais baixa ordem com os dois dígitos de mais alta ordem elevada ao quadrado é igual ao próprio número. Por exemplo, para o inteiro 3025, temos que:

$$30 + 25 = 55$$

$$55^2 = 3025$$

Salve o projeto com o nome “43\_soma\_dos\_digitos”.

44) (BACKES, 2012) Faça um programa que calcule a área de um triângulo, cuja base e altura são fornecidas pelo usuário. Esse programa não pode permitir a entrada de dados inválidos, ou seja, medidas menores ou iguais a 0. Salve o projeto com o nome “44\_area\_triangulo”.

45) (BACKES, 2012) Elabore um programa que faça leitura de vários números inteiros, até que se digite um número negativo. O programa tem que retornar o maior e o menor número lido. Salve o projeto com o nome "45\_maior\_e\_menor\_numeros".

46) (BACKES, 2012) Faça um programa que calcula a associação em paralelo de dois resistores R1 e R2 fornecidos pelo usuário via teclado. O programa fica pedindo estes valores e calculando até que o usuário entre com um valor para resistência igual a zero.

$$R = \frac{R1 * R2}{R1 + R2}$$

Salve o projeto com o nome "46\_resistores".

47) (BACKES, 2012) Faça um programa que leia um conjunto não determinado de valores, um de cada vez, e escreva para cada um dos valores lidos, o quadrado, o cubo e a raiz quadrada. Finalize a entrada de dados com um valor negativo ou zero. Salve o projeto com o nome "47\_calcular\_quadrado\_cubo\_raiz".

48) (BACKES, 2012) Faça um programa que leia um número indeterminado de idades de indivíduos (pare quando for informada a idade 0), e calcule a idade média desse grupo. Salve o projeto com o nome "48\_media\_idade".

49) (PUGA & RISSETI, 2016) Construa um algoritmo que encontre a mediana de uma sequência de números inteiros fornecida pelo usuário (número inicial e final), utilizando estrutura de repetição. Por exemplo, a mediana da sequência "1, 2, 3, 4, 5" é 3 e da sequência "2, 3, 4, 5, 6, 7, 8, 9" é  $(5 + 6) / 2 = 5,5$ . Como sugestão utilize a variável i para o número inicial e j para o número final, realizando operações de incremento e decremento. Salve o projeto com o nome "49\_mediana".

50) (PUGA & RISSETI, 2016) Uma empresa de recrutamento e seleção de funcionários possui, entre seus clientes, organizações em diversos ramos de atividade. Atende, em média, 30 candidatos por dia, número que aumenta quando a demanda cresce em decorrência de períodos de recessão. Para facilitar o trabalho de identificação do perfil dos candidatos que se inscrevem para as vagas, a empresa optou por fazer um programa de registrou de alguns dados, classificando as seguintes informações:

- a) O número de candidatos do sexo feminino.
- b) O número de candidatos do sexo masculino.
- c) A idade média dos homens com experiência.
- d) A percentagem dos homens entre 35 e 45 anos, entre o total de homens.
- e) A menor idade entre as mulheres que já têm experiência no serviço.
- f) O nível de escolaridade dos candidatos, considerando ensino fundamental, ensino médio, ensino superior e pós-graduação.

Faça um algoritmo para calcular e apresentar as informações mencionadas, sendo que, a cada iteração deve ser perguntado ao usuário se ele deseja cadastrar outro candidato, encerrando o programa, se a resposta for negativa. Salve o projeto com o nome `"50_selecao_funcionarios"`.

51) (PUGA & RISSETI, 2016) Escreva um algoritmo que leia uma quantidade qualquer de números, fornecidos pelo usuário. Faça a contagem e exiba quantos estão nos seguintes intervalos: [0 a 25.9], [26 a 50.9], [51 a 75.9] e [76 a 100], sendo que a entrada de dados deve terminar quando for digitado um número negativo. Salve o projeto com o nome `"51_intervalo_numeros"`.

52) (PUGA & RISSETI, 2016) Foi realizada uma pesquisa de algumas características físicas da população de uma região, que coletou os seguintes dados de cada habitante:

- a) Sexo (masculino e feminino).
- b) Cor dos olhos (azuis, verdes ou castanhos).
- c) Cor dos cabelos (loiros, ruivos, castanhos ou pretos).
- d) Idade.
- e) Altura.
- f) Peso.

Para que seja possível fazer um diagnóstico desta população, crie um algoritmo que calcule e apresente:

- a) A média da idade dos participantes.
- b) A média do peso e altura dos seus habitantes.
- c) A percentagem de pessoas do sexo feminino.
- d) A percentagem de pessoas do sexo masculino.
- e) Quantas pessoas possuem olhos verdes e cabelos louros.

Os resultados somente deverão ser apresentados quando o usuário informar que encerrou a entrada de dados. Salve o projeto com o nome `"52_censo"`.

53) (PUGA & RISSETI, 2016) Faça um algoritmo que, a partir de um número fornecido pelo usuário, inteiro e positivo, calcule e exiba seu fatorial ( $n!$ ). Salve o projeto com o nome `"53_fatorial"`.

54) (PUGA & RISSETI, 2016) Elabore um algoritmo que realize a potência de um número inteiro por outro, também inteiro e positivo, por meio de multiplicações sucessivas, sendo ambos informados pelo usuário. Salve o projeto com o nome `"53_potencia"`.

54) (BACKES, 2012) Faça um algoritmo que converta uma velocidade expressa em km/h para m/s e vice versa. Você deve criar um menu com as duas opções de conversão e com uma opção para finalizar o programa. O usuário poderá fazer quantas conversões desejar, sendo que o programa só será finalizado quando a opção de finalizar for escolhida. Salve o projeto com o nome `"54_conversao_velocidade"`.

55) (BACKES, 2012) Faça um programa que gera um número aleatório de 1 a 1000. O usuário deve tentar acertar qual o número foi gerado, a cada tentativa o programa deverá informar se o chute é menor ou maior que o número gerado. O programa acaba quando o usuário acerta o número gerado. O programa deve informar em quantas tentativas o número foi descoberto. Salve o projeto com o nome “55\_numeros\_aleatorios”.

56) (BACKES, 2012) Faça um programa que apresente um menu de opções para o cálculo das seguintes operações entre dois números:

- adição (opção 1)
- subtração (opção 2)
- multiplicação (opção 3)
- divisão (opção 4).
- saída (opção 5)

O programa deve possibilitar ao usuário a escolha da operação desejada, a exibição do resultado e a volta ao menu de opções. O programa só termina quando for escolhida a opção de saída (opção 5). Salve o projeto com o nome “56\_calculadora”.

57) (BACKES, 2012) O funcionário chamado Carlos tem um colega chamado João que recebe um salário que equivale a um terço do seu salário. Carlos gosta de fazer aplicações na caderneta de poupança e vai aplicar seu salário integralmente nela, pois está rendendo 2% ao mês. João aplicará seu salário integralmente no fundo de renda fixa, que está rendendo 5% ao mês. Construa um programa que deverá calcular e mostrar a quantidade de meses necessários para que o valor pertencente a João iguale ou ultrapasse o valor pertencente a Carlos. Teste com outros valores para as taxas. Salve o projeto com o nome “57\_aplicacao”.

58) (BACKES, 2012) Um funcionário recebe aumento anual. Em 1995 foi contratado por 200 reais. Em 1996 recebeu aumento de 1.5%. A partir de 1997, os aumentos sempre correspondem ao dobro do ano anterior. Faça programa que determine o salário atual do funcionário. Salve o projeto com o nome “58\_salario\_atual”.

60) (BACKES, 2012) Escreva um programa que receba como entrada o valor do saque realizado pelo cliente de um banco e retorne quantas notas de cada valor serão necessárias para atender ao saque com a menor quantidade de notas possível. Serão utilizadas notas de 100, 50, 20, 10, 5, 2 e 1 real. Salve o projeto com o nome “60\_saque\_banco”.

61) (BACKES, 2012) Escreva um programa que leia um número inteiro positivo n e em seguida imprima n linhas do chamado Triângulo de Floyd. Para n = 6, temos:

```
1
2 3
4 5 6
7 8 9 10
```



11 12 13 14 15  
16 17 18 19 20 21

Salve o projeto com o nome “61\_triangulo\_de\_floyd”.

62) (BACKES, 2012) Faça um programa que receba um número inteiro maior do que 1, e verifique se o número fornecido é primo ou não. Salve o projeto com o nome “62\_numero\_primo”.

63) (BACKES, 2012) Escreva um programa que leia um inteiro não negativo n e imprima a soma dos n primeiros números primos. Salve o projeto com o nome “63\_soma\_numeros\_primos”.

64) (BACKES, 2012) Faça um programa que conte quantos números primos existem entre a e b, onde a e b são números informados pelo usuário. Salve o projeto com o nome “64\_numeros\_primos\_entre\_a\_e\_b”.

65) (BACKES, 2012) Escreva um programa que leia o número de habitantes de uma determinada cidade, o valor do kwh, e para cada habitante entre com os seguintes dados: consumo do mês e o código do consumidor (1-Residencial, 2-Comercial, 3-Industrial). No final imprima o maior, o menor e a média do consumo dos habitantes; e por fim o total do consumo de cada categoria de consumidor. Salve o projeto com o nome “65\_calculo\_kwh”.

66) (BACKES, 2012) Faça um programa que calcule o maior número palíndromo feito a partir do produto de dois números de 3 dígitos. Ex: O maior palíndromo feito a partir do produto de dois números de dois dígitos é 9009 = 91\*99. Salve o projeto com o nome “66\_palindromo”.

67) (BACKES, 2012) Se os números de 1 a 5 são escritos em palavras: um, dois, três, quatro, cinco, então há  $2 + 4 + 4 + 6 + 5 = 22$  letras usadas no total. Faça um programa que conte quantas letras seriam utilizadas se todos os números de 1 a 1000 (mil) fossem escritos em palavras. OBS: Não conte espaços ou hifens. Salve o projeto com o nome “67\_palavras”.

68) (PUGA & RISSETI, 2016) Escreva um algoritmo que calcule o M.D.C. (Máximo Divisor Comum) entre A e B (números inteiros e positivos), sendo esses valores informados pelo usuário. Salve o projeto com o nome “68\_mdc”.

69) (PUGA & RISSETI, 2016) Um número perfeito é aquele igual à soma de seus divisores, como o  $6 = 3 + 2 + 1$ . Faça um algoritmo que verifique quais são os números perfeitos em um conjunto de números inteiros informado pelo usuário (inicial e final). Salve o projeto com o nome “69\_numero\_perfeito”.

70) (PUGA & RISSETI, 2016) Faça um algoritmo que mostre os conceitos finais de uma turma com 75 alunos, a partir das notas fornecidas pelo usuário, apresentando, para cada conceito, a quantidade de alunos e a média da nota verificada com base na tabela a seguir:

Faixa de nota	Conceito
De 0,0 a 2,9	E
De 3,0 a 4,9	D
De 5,0 a 6,9	C
De 7,0 a 8,9	B
De 9,0 a 10,0	A

Salve o projeto com o nome “70\_conceitos”.