Lista de exercícios Estrutura de Dados 1

# Nome: Edson G. M. Jordão- RGM: 32198914 Curso: Ciência da Computação - Turma: 2B edsongabrielmj@gmail.com

Universidade da Cidade de São Paulo (UNICID) - Rua Cesário Galeno, 448/475 São Paulo – SP – Brasil – CEP: 03071-000

***Abstract.*** *This project aims to answer the list of exercises in the subject of Data Structure I in C programming language, through knowledge obtained during classes, and our creativity and imagination when solving problems. With the help of complementary material provided by Professor Juliano Ratusznei.*

**Resumo.** Este trabalho tem por objetivo responder à lista de exercícios da matéria de Estrutura de Dados I na linguagem de programação C, através de conhecimentos obtidos durante as aulas, e nossa criatividade e imaginação na hora de resolver problemas. Com o auxílio do material complementar disponibilizado pelo professor Juliano Ratusznei.

São Paulo - 2023

# Descrição do Exercício 1 a ser solucionado.

1. Escreva um programa para escrever quando inicializado “Olá aluno hoje é sexta-feira”

# Resolução do Exercício 1 - Codificação:

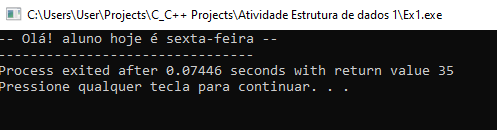
void main(void){

    setlocale(LC\_ALL,"Portuguese");

    printf("-- Olá aluno hoje é sexta-feira --");

}

**Execução do Exercício 1 – Resolução**



***Figura 1: tela da solução do exercício 1.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, relembrei os comandos básicos para se imprimir uma mensagem no terminal.

# Descrição do Exercício 2 a ser solucionado.

1. Dado a entrada via teclado do *nome de****usuário* *(seu nome)*** escrever na tela “Bem-vindo ***nome de usuário***”

# Resolução do Exercício 2 - Codificação:

void main(void){

    setlocale(LC\_ALL,"Portuguese");

    char nome[64];

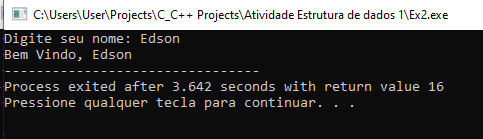
    printf("Digite seu nome: ");

    scanf("%s", nome);

    printf("Bem Vindo, %s", nome);

}

**Execução do Exercício 2 – Resolução**



***Figura 2: tela da solução do exercício 2.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, relembrei que um “String” em C, é um array/sequencia de caracteres, diferente de Java, com um tipo de dado dedicado.

# Descrição do Exercício 3 a ser solucionado.

1. Dado de entrada um ***numeral***escrever na tela “O número digitado foi: ***numeral***”

# Resolução do Exercício 3 - Codificação:

void main(void){

    setlocale(LC\_ALL,"Portuguese");

    int numeral;

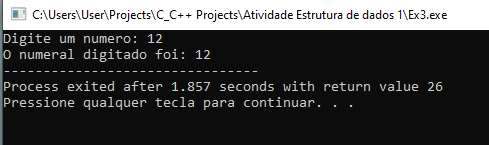
    printf("Digite um numero: ");

    scanf("%d", &numeral);

    printf("O numeral digitado foi: %d", numeral);

}

**Execução do Exercício 3 – Resolução**

***Figura 3: tela da solução do exercício 3.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, relembrei como se declara uma variável inteira e como se atribui um valor a ela quando não se atribui logo no início do programa.

# Descrição do Exercício 4 a ser solucionado.

1. Dado de entrada, um ***valor lógico*** escrever na tela “O valor oposto é: ***valor lógico***”

# Resolução do Exercício 4 - Codificação:

void main(void){

    setlocale(LC\_ALL,"Portuguese");

    bool polaridade;

    printf("Para definir a polaridade digite 1 ou 0: ");

    scanf("%d", &polaridade);

    printf("Valor digitado e: %d\n", polaridade);

    polaridade = inversor(polaridade);

    printf("O inverso e: %d", polaridade);

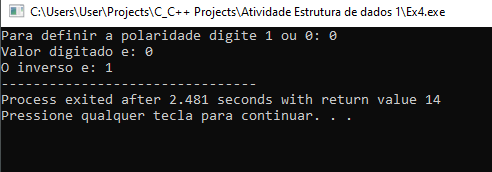
}

void inversor(polaridade){

    polaridade = !polaridade;

}

**Execução do Exercício 4 – Resolução**

***Figura 4: tela da solução do exercício 4.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, relembrei o funcionamento de função além da função principal(main), e como reatribuir um valor a uma variável.

# Descrição do Exercício 5 a ser solucionado.

1. Dado de entrada dois valores numéricos fazer a subtração e apresentar ao usuário.

# Resolução do Exercício 5 - Codificação:

int main(){

    setlocale(LC\_ALL,"Portuguese");

    int num1, num2, subtracao;

    printf("Digite dois números para subtrair: ");

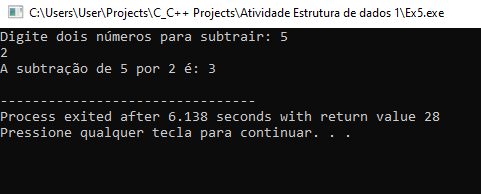
    scanf("%d %d", &num1, &num2);

    subtracao = num1 - num2;

    printf("A subtração de %d por %d é: %d\n", num1, num2, subtracao);

}

**Execução do Exercício 5 – Resolução**



***Figura 5: tela da solução do exercício 5.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, relembrei a forma com que se faz uma codificação de uma equação e seu funcionamento.

# Descrição do Exercício 6 a ser solucionado.

1. Escreva um programa que imprima os 10 primeiros números negativos.

# Resolução do Exercício 6 - Codificação:

void main(void){

    setlocale(LC\_ALL, "Portuguese");

    int contador = 0;

    printf("Os dex primeiros números negativos são: \n");

    while (contador > -11) {

        printf("%d ", contador);

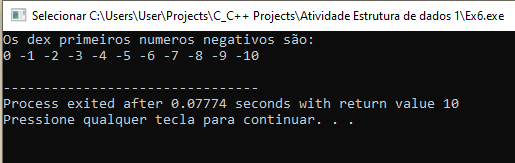
        contador--;

    }

    printf("\n");

}

**Execução do Exercício 6 – Resolução**

***Figura 6: tela da solução do exercício 6.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi a importância de um laço de repetição chamado while e o seu funcionamento, na prática.

# Descrição do Exercício 7 a ser solucionado.

1. Dado um número verificar se ele é maior que 10.

# Resolução do Exercício 7 - Codificação:

void main(void){

    setlocale(LC\_ALL, "Portuguese");

    int numero;

    printf("Digite um número positivo: ");

    scanf("%d", &numero);

    if (numero > 10) {

        printf("%d é maior que 10.\n", numero);

    }else if (numero < 0) {

        printf("%d Burro tem que ser positivo.\n", numero);

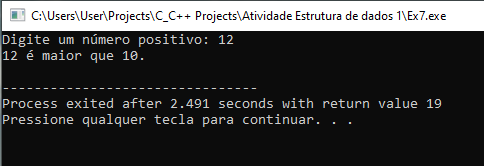
    }else {

        printf("%d não é maior que 10.\n", numero);

    }

}

**Execução do Exercício 7 – Resolução**

***Figura 7: tela da solução do exercício 7.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi a importância da estrutura if, else if e else, além de ter mais conhecimento sobre o seu funcionamento.

# Descrição do Exercício 8 a ser solucionado.

1. Dado um número, verificar se ele é positivo ou negativo.

# Resolução do Exercício 8 - Codificação:

void main(void){

    setlocale(LC\_ALL, "Portuguese");

    int numero;

    printf("Digite um nÚmero: ");

    scanf("%d", &numero);

    if (numero > 0) {

        printf("%d é positivo.\n", numero);

    }else if (numero < 0) {

        printf("%d é negativo.\n", numero);

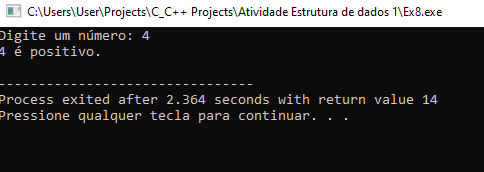
    }else {

        printf("O número é zero.\n");

    }

}

**Execução do Exercício 8 – Resolução**

***Figura 8: tela da solução do exercício 8.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi a importância dos operadores lógicos em um código.

# Descrição do Exercício 9 a ser solucionado.

1. Dados dois números diferentes, dividir o maior pelo menor valor.

# Resolução do Exercício 9 - Codificação:

int main(void){

    setlocale(LC\_ALL, "Portuguese");

    int maior, menor, num1, num2;

    printf("Digite dois números diferentes: ");

    scanf("%d %d", &num1, &num2);

    if (num1 > num2) {

        maior = num1;

        menor = num2;

    } else {

        maior = num2;

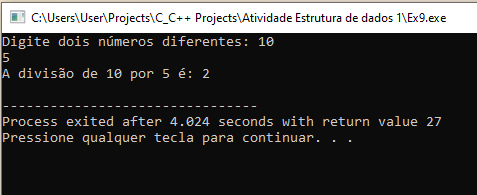
        menor = num1;

    }

    printf("A divisão de %d por %d é: %d\n", maior, menor, maior/menor);

}

**Execução do Exercício 9 – Resolução**



***Figura 9: tela da solução do exercício 9.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi melhor o funcionamento da atribuição de valores em uma estrutura condicional simples.

# Descrição do Exercício 10 a ser solucionado.

1. Dado a formula **Y = X + 5** escrever o valor de **Y** quando dado uma entrada de **X**.

# Resolução do Exercício 10 - Codificação:

int main(void){

    setlocale(LC\_ALL, "Portuguese");

    int x, y;

    printf("Digite um valor para X: ");

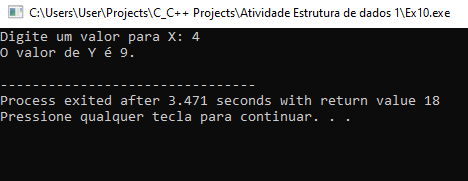
    scanf("%d", &x);

    y = x + 5;

    printf("O valor de Y é %d.\n", y);

}

**Execução do Exercício 10 – Resolução**



***Figura 10: tela da solução do exercício 10.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi outra maneira de fazer uma conta aritmética, atribuindo a uma variável antes de imprimir.

# Descrição do Exercício 11 a ser solucionado.

1. Dado a fórmula **media = n1+n2+n3/3** escrever o valor do **resultado** ao usuário quando ele digitar o valor de **n1, n2** e **n3**.

# Resolução do Exercício 11 - Codificação:

int main(void){

    setlocale(LC\_ALL, "Portuguese");

    float n1, n2, n3, media;

    printf("Digite três valores para n1, n2 e n3: ");

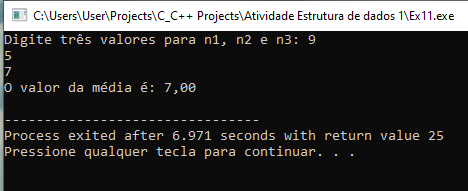
    scanf("%f %f %f", &n1, &n2, &n3);

    media = (n1 + n2 + n3) / 3;

    printf("O valor da média é: %.2f\n", media);

}

**Execução do Exercício 11 – Resolução**



***Figura 11: tela da solução do exercício 11.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi a importância da declaração de variáveis do tipo **float**, pois se fossem declaradas com o tipo inteiro e se o valor do divisor for par e do dividendo um número impar ou visse e versa o código estaria errado e nem executaria.

# Descrição do Exercício 12 a ser solucionado.

1. Dado a fórmula **A/5 = B/(A+2)** calcular o valor de **B** dado uma entrada de **A**.

# Resolução do Exercício 12 - Codificação:

int main(void){

    setlocale(LC\_ALL, "Portuguese");

    float a, b;

    printf("Digite o valor de a: ");

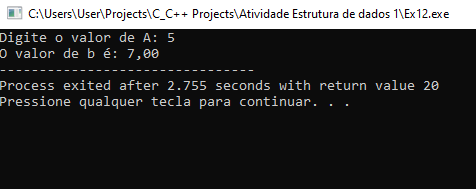
    scanf("%f", &a);

    b = (a \* a + 2 \* a) / 5.0;

    printf("O valor de b é: %.2f", b);

}

**Execução do Exercício 12 – Resolução**



***Figura 12: tela da solução do exercício 12.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi a importância do raciocínio lógico matemático para resolver questões do dia a dia.

# Descrição do Exercício 13 a ser solucionado.

1. Escreva um programa que dados 15 números, imprima seus quadrados, obs: você não precisa necessariamente imprimir no final.

# Resolução do Exercício 13 - Codificação:

int main(void){

    setlocale(LC\_ALL, "Portuguese");

    int num, i, quadrado;

    for (i = 1; i <= 15; i++) {

        printf("Digite o %dº número: ", i);

        scanf("%d", &num);

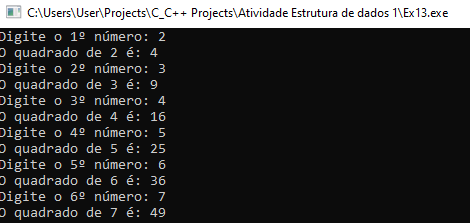
        quadrado = num \* num;

        printf("O quadrado de %d é: %d\n", num, quadrado);

    }

}

**Execução do Exercício 13 – Resolução**



***Figura 13: tela da solução do exercício 13.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi a importância de um laço de repetição chamado for que por padão recebe três cláusulas, uma variável com um valor inicializado, uma condição lógica e um passo ou uma ação.

# Descrição do Exercício 14 a ser solucionado.

1. Escreva um programa que dado o primeiro número e a razão de uma sequência, imprima seus dez primeiros termos: você não precisa necessariamente imprimir no final.

# Resolução do Exercício 14 - Codificação:

int main(void){

    setlocale(LC\_ALL, "Portuguese");

    int primeiro, razao, i, termo;

    printf("Digite o primeiro número e a razão da sequência: ");

    scanf("%d %d", &primeiro, &razao);

    for (i = 0; i <= 10; i++) {

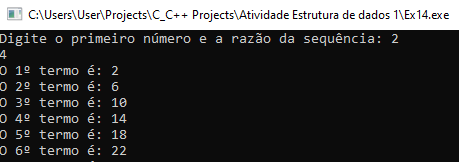
        termo = primeiro + i \* razao;

        printf("O %dº termo é: %d\n", i + 1, termo);

    }

}

**Execução do Exercício 14 – Resolução**

***Figura 14: tela da solução do exercício 14.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, compreendi melhor o funcionamento de um laço de repetição,.

# Descrição do Exercício 15 a ser solucionado.

1. Escreva um programa que imprima na tela o fatorial de um programa recebido

Ex:   5! = 5 x 4 x 3 x 2 x 1

Ex:   0! = 1

# Resolução do Exercício 15 - Codificação:

int main(void){

    setlocale(LC\_ALL, "Portuguese");

    int numero, i, fatorial = 1;

    printf("Digite um número inteiro: ");

    scanf("%d", &numero);

    if (numero == 0) {

        printf("0! = 1");

    }

    for (i = 1; i <= numero; i++) {

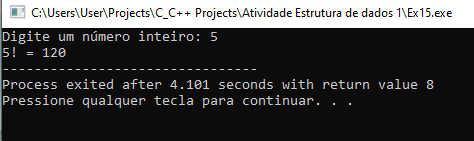
        fatorial \*= i;

    }

    printf("%d! = %d", numero, fatorial);

}

**Execução do Exercício 15 – Resolução**

***Figura 15: tela da solução do exercício 15.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi o funcionamento de uma atribuição de valores de uma forma simplificada.

# Descrição do Exercício 16 a ser solucionado.

1. Escreva um programa que dado um número, ele diz se é um número primo ou não.

# Resolução do Exercício 16 - Codificação:

int main(void){

  setlocale(LC\_ALL, "Portuguese");

  int num, i, primo = 1;

  printf("Digite um número inteiro: ");

  scanf("%d", &num);

// Comentario para considerar o numero 1 primo.

//  if (num < 2) {

//    primo = 0;

//  }

  for (i = 2; i <= num / 2; i++) {

    if (num % i == 0) {

      primo = 0;

    }

  }

  if (primo) {

    printf("%d é um número primo.", num);

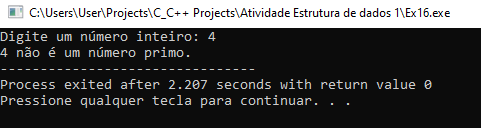
  } else {

    printf("%d não é um número primo.", num);

  }

}

**Execução do Exercício 16 – Resolução**



***Figura 16: tela da solução do exercício 16.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi a importância da mescla entre duas estruturas diferentes, o laço de repetição e a condicional para desenvolver uma abordagem diferente e solucionar o problema de uma forma sucinta.

# Descrição do Exercício 17 a ser solucionado.

1. Escreva um programa que imprime todos os números primos positivos a partir do zero até um número digitado.

# Resolução do Exercício 17 - Codificação:

int main(void){

  setlocale(LC\_ALL, "Portuguese");

  int num, i, j, primo;

  printf("Digite um número inteiro: ");

  scanf("%d", &num);

  printf("Os números primos até %d são:\n 1 ", num);

  for (i = 2; i <= num; i++) {

    primo = 1;

    for (j = 2; j <= i / 2; j++) {

      if (i % j == 0) {

        primo = 0;

      }

    }

    if (primo) {

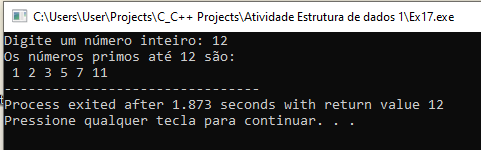
      printf("%d ", i);

    }

  }

}

**Execução do Exercício 17 – Resolução**

***Figura 17: tela da solução do exercício 17.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi melhor o comportamento geral entre dois laços de repetição.

# Descrição do Exercício 18 a ser solucionado.

1. Escreva um programa que imprima o seguinte vetor:

1   1   2   3   5   8   13   21   34   55   86   144   233   300   533

# Resolução do Exercício 18 - Codificação:

int main(void){

  setlocale(LC\_ALL, "Portuguese");

  int vetor[15], i;

  vetor[0] = 1;

  vetor[1] = 1;

  for (i = 2; i < 15; i++) {

    vetor[i] = vetor[i-1] + vetor[i-2];

  }

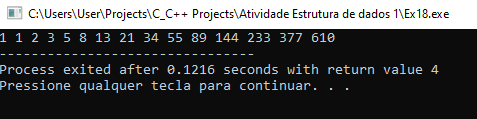
  for (i = 0; i < 15; i++) {

    printf("%d ", vetor[i]);

  }

}

**Execução do Exercício 18 – Resolução**



***Figura 18: tela da solução do exercício 18.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi a importância de desenvolver laços de repetição para criar listas de uma forma automática.

# Descrição do Exercício 19 a ser solucionado.

1. Escreva um programa que imprima a seguinte matriz:

10 11 12 13 14

15 16 17 18 19

20 21 22 23 24

25 26 27 28 29

30 31 32 33 34

# Resolução do Exercício 19 - Codificação:

int main(void){

  setlocale(LC\_ALL, "Portuguese");

  int matriz[5][5], i, j, valor = 10;

  for (i = 0; i < 5; i++) {

    for (j = 0; j < 5; j++) {

      matriz[i][j] = valor;

      valor++;

    }

  }

  for (i = 0; i < 5; i++) {

    for (j = 0; j < 5; j++) {

      printf("%d ", matriz[i][j]);

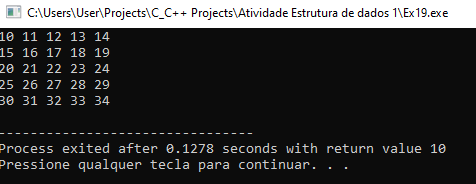
    }

    printf("\n");

  }

}

**Execução do Exercício 19 – Resolução**



***Figura 19: tela da solução do exercício 19.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi a importância de desenvolver laços de repetição criando listas em colunas(matriz) que posteriormente podem ser utilizadas para fins gerais na computação.

# Descrição do Exercício 20 a ser solucionado.

1. Escreva um programa que imprime os seguintes dados na tela

1-1 1-2 1-3 1-4 1-5

2-1 2-2 2-3 2-4 2-5

3-1 3-2 3-3 3-4 3-5

4-1 4-2 4-3 4-4 4-5

5-1 5-2 5-3 5-4 5-5

# Resolução do Exercício 20 - Codificação:

int main(void){

  setlocale(LC\_ALL, "Portuguese");

  int i, j;

  for (i = 1; i <= 5; i++) {

    for (j = 1; j <= 5; j++) {

      printf("%d-%d ", i, j);

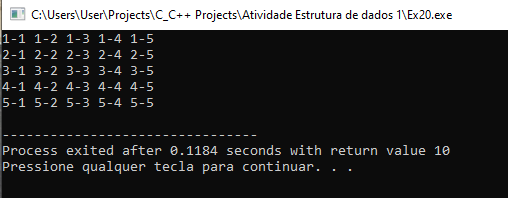
    }

    printf("\n");

  }

}

**Execução do Exercício 20 – Resolução**



***Figura 20: tela da solução do exercício 20.***

# Descrição da aprendizagem obtida através da problemática:

Com esse exercício, entendi melhor o conceito de uma matriz, além de ter uma visão mais ampla do funcionamento de laços encadeados.

# Referências:

ASCENCIO, A. F. G.; ARAÚJO, G. S. de. Estruturas de dados: algoritmos, análise da complexidade e implementações em Java e C/C. São Paulo: Pearson Prentice Hall, 2010, pág. 176 a 188. (e-book). Disponível em: <https://github.com/free-educa/books/blob/main/books/Estrutura%20de%20Dados%2C%20algoritmos%2C%20an%C3%A1lise%20da%20complexidade%20e%20implementa%C3%A7%C3%B5es%20em%20Java%20e%20C%2C%20C--.pdf>. Acessado em mar/2023.

ACADEMY CODE; Learning C programming language. Code Academy, disponível em: <https://www.codecademy.com/courses/learn-c/lessons> Acessado em abr/2022.

HASTE, JOHN; Curso de Linguagem C para Iniciantes. One Day Code, disponível em: <https://www.youtube.com/watch?v=MHmXIpeF-cQ&list=PLGgRtySq3SDMLV8ee7p-rA9y032AU3zT8>. Acessado em mar/2023.