**CADERNO DE RESPOSTAS DA ATIVIDADE PRÁTICA DE:**

**Linguagem de Programação**

**ALUNO: Vinícius Valle Beraldo – 4180020**

**ANO**

**2024**



Prof. Winston Sen Lun Fung, Esp.

|  |
| --- |
| **PRÁTICA 01** |
|  |
| **Escreva um algoritmo em linguagem C que atenda os seguintes requisitos:**   * **Crie um registro para armazenar o seu Nome Completo e o seu RU.** * **Solicite ao usuário que digite o seu RU e o seu Nome Completo.** * **Crie um ponteiro para o registro.** * **Através do ponteiro para o registro verifique se o RU digitado é um número par ou ímpar e informe a resposta na tela.** * **Através do ponteiro para o registro imprima o Nome Completo informado de traz para frente, invertido. Por exemplo:**   **Pedro Alvares Cabral -> larbaC seravlA ordeP**  **Para demonstrar o funcionamento faça as capturas de tela do terminal utilizando seu nome completo e o seu RU.** |
| 1. **Apresentação do código-fonte:**   #include <stdio.h>  #include <string.h>  //Registro  typedef struct  {  char nome[100];  int ru;  } Register;  int main()  {  Register aluno;  // passa os valores para variáveis  printf("Digite o seu RU: ");  scanf("%d", &aluno.ru);  printf("Digite o seu nome completo: ");  scanf(" *%*[^\n]", aluno.nome);  // ponteiros  Register \*pAluno = &aluno;  if (pAluno->ru % 2 == 0)  {  printf("O RU %d é par.\n", pAluno->ru);  }  else  {  printf("O RU %d é impar.\n", pAluno->ru);  }  printf("O seu nome invertido:");  for (int i = strlen(pAluno->nome) - 1; i >= 0; i--)  {  printf("%c", pAluno->nome[i]);  }  printf("\n");  return 0;  } |
| 1. **Apresentar a captura de tela evidenciando o funcionamento:** |

|  |
| --- |
| **PRÁTICA 02** |
|  |
| **Escreva um algoritmo em linguagem C que atenda os seguintes requisitos:**   * **Crie um vetor com a quantidade de dígitos do seu RU.** * **Solicite que usuário digite o seu RU, cada digito digitado deve ser armazenado em uma posição do vetor.** * **Utilizando ponteiros verifique qual o maior e menor número contido neste vetor e imprima na tela.**   **Para demonstrar o funcionamento faça as capturas de tela do terminal utilizando seu RU.** |
| 1. **Apresentação do código-fonte:**   #include <stdio.h>  int main(){  // 4180020  int ru[7];  int \*bigger = ru;  int \*lower = ru;  char str\_ru[7];  //armazena os ru em uma variável  printf("Digite o seu RU: ");  scanf("%7s", str\_ru);  //pega essa variável com o ru e distruibui em cada um dos array com seu index  for(int i = 0; i < 7; i++){  ru[i] = str\_ru[i] - '0';  }  // loop para verificar se ele é maior ou menor número do vetor  for(int i = 1; i < 7; i++){  if(ru[i] > \*bigger){  bigger = ru + i;  }else{  lower = ru + i;  }  }  //resultado no console  printf("O maior é %d\n",\*bigger);  printf("O menor é %d\n", \*lower);  return 0;  } |
| 1. **Apresentar a captura de tela evidenciando o funcionamento:** |

|  |
| --- |
| **PRÁTICA 03** |
|  |
| **Faça um programa, em linguagem C, para calcular a área e o perímetro de um hexágono. O programa deve implementar uma função chamada calc\_hexa que calcula a área e o perímetro de um hexágono regular de lado L.**  **O programa deve solicitar ao usuário o lado do polígono, calcular e imprimir a área e o perímetro do polígono.**  **O programa termina quando for digitado um valor negativo qualquer para o lado. A função deve obedecer ao seguinte protótipo:**  **void calc\_hexa(float l, floar \*area, float \*perimetro);**  **Lembrando que a área e o perímetro de um hexágono regular são dados por:**  **Para demonstrar o funcionamento faça os print de tela utilizando o primeiro número do seu RU.** |
| 1. **Apresentação do código-fonte:**   #include <stdio.h>  #include <stdlib.h>  #include <math.h>  void calc\_hexa(float l, float \*area, float \*perimetro){  \*area = (3\*M\_PI\*l\*l)/2;  \*perimetro = 6 \* l;  }  int main(){  float lado, area, perimetro;  printf("Digite o lado do hexágono (valor negativo para sair):");  scanf ("%f", &lado);  while(lado >= 0){  calc\_hexa(lado, &area, &perimetro);  printf("Área do hexágono: %.2f\n", area);  printf("Perímetro do hexágono: %.2f\n", perimetro);  printf("Digite o lado do hexágono (valor negativo para sair):");  scanf ("%f", &lado);  }  return 0;  } |
| 1. **Apresentar a captura de tela evidenciando o funcionamento:** |

|  |
| --- |
| **PRÁTICA 04** |
|  |
| **Faça, em linguagem C, uma função recursiva para o cálculo potenciação. Sabe-se que o cálculo pode ser feito através de multiplicações sucessivas.**  **Para demonstrar o funcionamento utilize o primeiro digito do seu RU como base e o último digito como expoente.** |
| 1. **Apresentação do código-fonte:**   #include <stdio.h>  int potencia(int base, int exp){  if(exp == 0) {return 1; }  return base \* potencia(base, exp - 1);  }  int main() {  // ru 1480020  int base = 1; // primeiro dígito do RU  int exp = 0; // último dígito do RU  int result = potencia(base, exp);  printf("%d elevado a %d: %d\n", base, exp, result);  } |
| 1. **Apresentar a captura de tela evidenciando o funcionamento:** |

|  |
| --- |
| **PRÁTICA 05** |
|  |
| **Crie um programa, em linguagem C, que receba 10 registros contendo, Nome, RU, telefone email. Solicite que sejam digitados todos os dados de todos os registros e ao final salve-os em um arquivo.csv, utilize o ; (ponto e vírgula) para separador e campo. O nome do arquivo deve ser o seu número de RU [N°\_DO\_RU.CSV].**  **ATENÇÃO: ASSISTA OS VIDEOS DA AULA 8 – MATERIAL COMPLEMENTAR.** |
| 1. **Apresentação do código-fonte:**   #include <stdio.h>  #include <string.h>  #include <stdlib.h>  typedef struct {  char name[50];  char ru[10];  char telefone[15];  char email[50];  } Register;  int main(){  Register registros[10];  FILE \*file;  char fileName[] = "1480020.csv"; // nome do arquivo igual meu RU  for (int i = 0; i < 10; i++){  printf("Registro %d:\n", i + 1);  printf("Nome: ");  fgets(registros[i].name, 50, stdin);  registros[i].name[strcspn(registros[i].name, "\n")] = 0; // remove nova linha  printf("RU: ");  fgets(registros[i].ru, 10, stdin);  registros[i].ru[strcspn(registros[i].ru, "\n")] = 0;  printf("Telefone: ");  fgets(registros[i].telefone, 15, stdin);  registros[i].telefone[strcspn(registros[i].telefone, "\n")] = 0;  printf("Email: ");  fgets(registros[i].email, 50, stdin);  registros[i].email[strcspn(registros[i].email, "\n")] = 0;  }  file = fopen(fileName, "w");  if(file == NULL){  printf("Erro ao abrir\n");  return 1;  }  fprintf(file, "Nome;RU;Telefone;Email\n");  for(int i = 0; i < 10; i++){  fprintf(file, "%s;%s;%s;%s\n",registros[i].name, registros[i].ru, registros[i].telefone, registros[i].email);  }  fclose(file);  printf("Dados salvos no arquivo %s com sucesso!\n", fileName);  return 0;  } |
| 1. **Apresentar a captura de tela evidenciando o funcionamento:** |
| 1. **Apresentar a captura de tela do arquivo CSV:** |