



CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
ESCOLA SUPERIOR POLITÉCNICA
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE LINGUAGEM DE PROGRAMAÇÃO

Atividade Prática

Autor: Paulo Vinicius Pereira Pinheiro

No. RU: 3760288

Prof. Sandro de Araújo

Prof. Winston Sen Lun Fung

Juazeiro do Norte - Ceará

2022

LISTA DE ILUSTRAÇÕES

Figura 1 – Tela inicial do programa.	7
Figura 2 – Telas com os campos de inserção de dados.	7
Figura 3 – Outras telas complementares com os campos de inserção de dados. . .	7
Figura 4 – Tela final com a segregação dos dados por mês.	8
Figura 5 – Tela final com a segregação dos dados por mês.	11
Figura 6 – Tela inicial com a solicitação do tipo do cômodo.	14
Figura 7 – Tela final com a apresentação dos resultados.	15
Figura 8 – Tela final com a apresentação dos resultados.	15
Figura 9 – Tela com a solicitação do vetor de entrada e resultados.	18
Figura 10 – Tela com a solicitação dos registros de entrada.	21
Figura 11 – Tela com a criação do arquivo .csv e a inscrição dos dados de input. . .	21

LISTINGS

1.1	Código fonte da prática 1 - registro de pessoas e aniversários.	5
2.1	Código fonte da prática 2 - verificação e contagem de caracteres.	9
3.1	Código fonte da prática 3 - adequação do número de lâmpadas a comodors. . .	12
4.1	Código fonte da prática 4 - verificação do menor e maior dígitos.	16
5.1	Código fonte da prática 5 - armazena dados em arquivo .csv.	19

SUMÁRIO

	Listings	2
	INTRODUÇÃO	4
1	PRÁTICA 1	5
1.1	ENUNCIADO	5
1.2	CÓDIGO-FONTE DA SOLUÇÃO	5
1.3	DISCUSSÃO DOS RESULTADOS	7
2	PRÁTICA 2	9
2.1	ENUNCIADO	9
2.2	CÓDIGO-FONTE DA SOLUÇÃO	9
2.3	DISCUSSÃO DOS RESULTADOS	11
3	PRÁTICA 3	12
3.1	ENUNCIADO	12
3.2	CÓDIGO-FONTE DA SOLUÇÃO	12
3.3	DISCUSSÃO DOS RESULTADOS	14
4	PRÁTICA 4	16
4.1	ENUNCIADO	16
4.2	CÓDIGO-FONTE DA SOLUÇÃO	16
4.3	DISCUSSÃO DOS RESULTADOS	18
5	PRÁTICA 5	19
5.1	ENUNCIADO	19
5.2	CÓDIGO-FONTE DA SOLUÇÃO	19
5.3	DISCUSSÃO DOS RESULTADOS	21
	CONCLUSÃO	22

INTRODUÇÃO

A linguagem de programação C foi desenvolvida por Dennis Ritchie na década de 70 como um aprimoramento da linguagem B. No entanto, somente em 1983, o ANSI (*AMERICAN NATIONAL STANDARD INSTITUTE*) publicou a primeira versão da linguagem C padronizada para sanar as discrepâncias encontradas na linguagem.

Com a popularidade dos microcomputadores, logo a linguagem se tornou universalmente utilizada. Inúmeros foram os compiladores criados para a linguagem C, que foram usados em diversos sistemas operacionais.

C frequentemente é chamada de linguagem de médio nível, pois é uma linguagem de baixo nível, com menos recursos e mais pouco espaço de memória. Isso não significa que ela seja menos poderosa que as demais, como PASCAL e BASIC. Tampouco se deve comparar a mesma com ASSEMBLY. Ela é conhecida da forma mencionada acima pois engloba recursos de linguagens de baixo nível e de alto nível.

Este trabalho é fruto do desenvolvimento da atividade prática solicitada na disciplina de Linguagem de programação. O mesmo está dividido em cinco partes, cada uma correspondendo a uma prática solicitada. Tal formato difere do tradicional modelo de relatório pois tenta simplificar a forma de apresentação dos resultados. E não custa enfatizar que o mesmo segue o modelo solicitado pelo professor.

1 PRÁTICA

1.1 ENUNCIADO

Escreva um algoritmo em linguagem C que atenda aos seguintes requisitos:

- Os campos de um registro devem armazenar o Nome, dia e mês de aniversário.
- Solicite ao usuário que digite 08 registros.
- Os registros devem ser armazenados em um vetor.
- Através do ponteiro para o vetor de registro mostre em cada um dos meses do ano quem são as pessoas que fazem aniversário.

Para demonstrar o funcionamento faça as capturas de tela do terminal utilizando seu nome completo e o seu dia e mês de aniversário em um dos registros de entrada solicitado.

1.2 CÓDIGO-FONTE DA SOLUÇÃO

Para o desenvolvimento da solução foi utilizada a versão 2022 do Microsoft Visual Studio. Abaixo, está o código-fonte da solução.

Listing 1.1 – Código fonte da prática 1 - registro de pessoas e aniversários.

```

1 /*
2 UNINTER - ENGENHARIA DA COMPUTACA
3 Disciplina de Linguagem de Programacao
4 Exercicio 1 - Atividade pratica
5
6 Autor: Paulo Vinicius Pereira Pinheiro
7 RU: 3760288
8 */
9
10 #include <stdlib.h>
11 #include <stdio.h>
12
13 void titulo();
14
15 struct dados
16 {
17     char nome[50];
18     int dia;
19     int mes;
20 };
21 typedef struct dados;
22 void titulo() {
23     printf("##### Exercicio 1 - Atividade Pratica
        #####\n");

```

```
24 printf("Aluno: Paulo Vinicius P. Pinheiro\n");
25 printf("RU: 3760288\n\n");
26 };
27
28 int main()
29 {
30     const int length = 8;
31     const int m = 12;
32     char c;
33     int j = 0, k = 0;
34
35     dados pessoas[length][{}], *p_pessoas;
36     printf("##### Registro de Funcionarios
37     #####\n\n");
38     // Etapa de coleta de dados
39     for (int i = 0; i < length; i++)
40     {
41         titulo();
42         printf("Registro no. %d\n\n", i+1);
43
44         printf("Digite o nome: ");
45         gets_s(pessoas[i].nome, 49);
46
47         printf("\nDigite o dia do aniversario: ");
48         scanf_s("%d", &pessoas[i].dia);
49         c = getchar();
50
51         printf("\nDigite o mes do aniversario: ");
52         scanf_s("%d", &pessoas[i].mes);
53         c = getchar();
54
55         system("cls");
56     }
57     // Etapa de apresentacao dos aniversariantes do mes
58     titulo();
59     p_pessoas = &pessoas[0];
60     printf("Lista de aniversariantes do mes\n");
61     printf("Numero de registros: %d\n\n", length);
62
63     for (j = 1; j < m+1; j++)
64     {
65         printf("\nAniversariantes do mes %d\n", j);
66         while (k < 8)
67         {
68             if (p_pessoas[k].mes == j)
69             {
70                 printf("%s, no dia %d.\n", p_pessoas[k].nome, p_pessoas[k].dia);
71             }
72             k++;
73         }
74         k = 0;
75     }
76 }
```

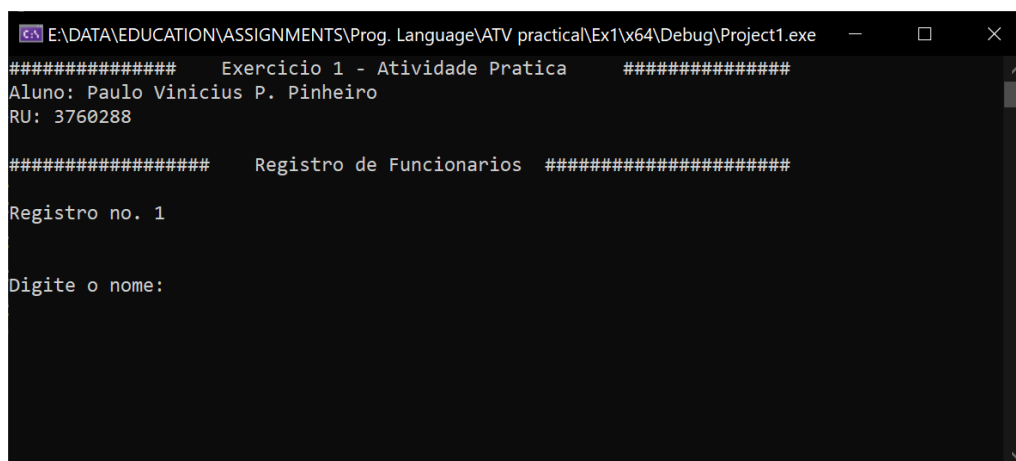
O código apresentado está dividido em duas partes: a primeira realiza a aquisição dos registros e suas datas de aniversário. Nesta parte, através da instrução de sistema `system("cls")`, o sistema limpa a tela após cada inserção de dados. Algumas telas são apresentadas na seção de discussão dos resultados.

A segunda parte, responsável pela seleção e exibição dos registros, utiliza um vetor de ponteiros na implementação.

1.3 DISCUSSÃO DOS RESULTADOS

Inicia-se a apresentação dos resultados apresentando a tela inicial do programa no terminal, como mostra a Figura 1.

Figura 1 – Tela inicial do programa.



```
E:\DATA\EDUCATION\ASSIGNMENTS\Prog. Language\ATV practical\Ex1\64\Debug\Project1.exe
##### Exercício 1 - Atividade Pratica #####
Aluno: Paulo Vinicius P. Pinheiro
RU: 3760288

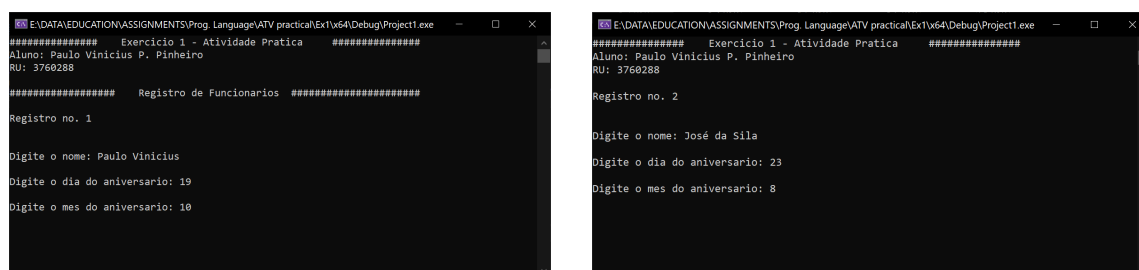
##### Registro de Funcionarios #####

Registro no. 1

Digite o nome:
```

O título incluído no programa é apresentado em todas as telas. Conforme elencado anteriormente, a aquisição dos registros é realizado em tela individual. Acima da solicitação do nome, percebe-se que o usuário pode identificar qual registro está sendo inserido. Nas Figuras 2 e 3 são apresentadas algumas das telas de inserção de dados.

Figura 2 – Telas com os campos de inserção de dados.



```
E:\DATA\EDUCATION\ASSIGNMENTS\Prog. Language\ATV practical\Ex1\64\Debug\Project1.exe
##### Exercício 1 - Atividade Pratica #####
Aluno: Paulo Vinicius P. Pinheiro
RU: 3760288

##### Registro de Funcionarios #####

Registro no. 1

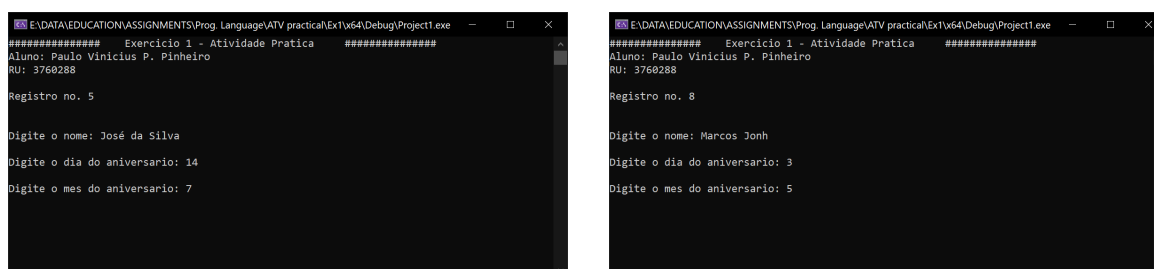
Digite o nome: Paulo Vinicius
Digite o dia do aniversario: 19
Digite o mes do aniversario: 10

E:\DATA\EDUCATION\ASSIGNMENTS\Prog. Language\ATV practical\Ex1\64\Debug\Project1.exe
##### Exercício 1 - Atividade Pratica #####
Aluno: Paulo Vinicius P. Pinheiro
RU: 3760288

Registro no. 2

Digite o nome: José da Sila
Digite o dia do aniversario: 23
Digite o mes do aniversario: 8
```

Figura 3 – Outras telas complementares com os campos de inserção de dados.



```
E:\DATA\EDUCATION\ASSIGNMENTS\Prog. Language\ATV practical\Ex1\64\Debug\Project1.exe
##### Exercício 1 - Atividade Pratica #####
Aluno: Paulo Vinicius P. Pinheiro
RU: 3760288

Registro no. 5

Digite o nome: José da Silva
Digite o dia do aniversario: 14
Digite o mes do aniversario: 7

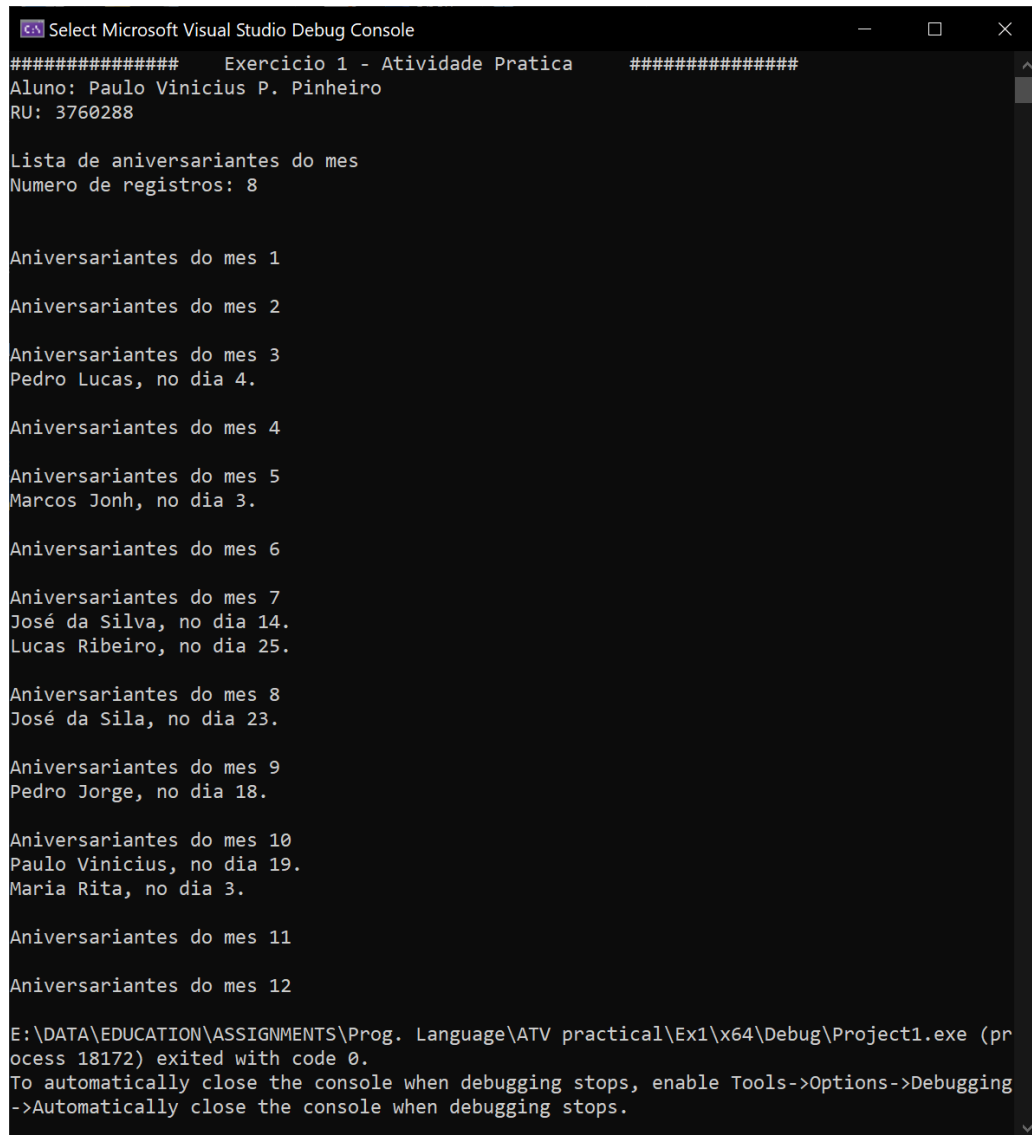
E:\DATA\EDUCATION\ASSIGNMENTS\Prog. Language\ATV practical\Ex1\64\Debug\Project1.exe
##### Exercício 1 - Atividade Pratica #####
Aluno: Paulo Vinicius P. Pinheiro
RU: 3760288

Registro no. 8

Digite o nome: Marcos Jonh
Digite o dia do aniversario: 3
Digite o mes do aniversario: 5
```


Na figura 4 está a tela final com a captura total dos dados e sua separação pelos meses de aniversário. Além do solicitado, foi incluído também no resultado final o dia em que o funcionário tem seu aniversário.

Figura 4 – Tela final com a segregação dos dados por mês.



```
Select Microsoft Visual Studio Debug Console
##### Exercício 1 - Atividade Pratica #####
Aluno: Paulo Vinicius P. Pinheiro
RU: 3760288

Lista de aniversariantes do mes
Numero de registros: 8

Aniversariantes do mes 1
Aniversariantes do mes 2
Aniversariantes do mes 3
Pedro Lucas, no dia 4.
Aniversariantes do mes 4
Aniversariantes do mes 5
Marcos Jonh, no dia 3.
Aniversariantes do mes 6
Aniversariantes do mes 7
José da Silva, no dia 14.
Lucas Ribeiro, no dia 25.
Aniversariantes do mes 8
José da Sila, no dia 23.
Aniversariantes do mes 9
Pedro Jorge, no dia 18.
Aniversariantes do mes 10
Paulo Vinicius, no dia 19.
Maria Rita, no dia 3.
Aniversariantes do mes 11
Aniversariantes do mes 12

E:\DATA\EDUCATION\ASSIGNMENTS\Prog. Language\ATV practical\Ex1\x64\Debug\Project1.exe (process 18172) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging
->Automatically close the console when debugging stops.
```

2 PRÁTICA

2.1 ENUNCIADO

Faça um programa onde o usuário digita 3 informações a respeito de uma pessoa: Nome, endereço e telefone. Concatene essas três informações em uma única string e faça uma contagem de quantas letras do alfabeto estão presentes nesta string (considerando as redundâncias) e de dígitos numéricos. Os espaços e os caracteres de pontuação devem ser ignorados (as funções de contagem já fazem isso).

2.2 CÓDIGO-FONTE DA SOLUÇÃO

O código fonte desenvolvido como solução para o problema do enunciado está abaixo:

Listing 2.1 – Código fonte da prática 2 - verificação e contagem de caracteres.

```
1 /*
2 UNINTER - ENGENHARIA DA COMPUTAÇÃO
3 Disciplina de Linguagem de Programação
4 Exercício 2 - Atividade prática
5
6 Autor: Paulo Vinicius Pereira Pinheiro
7 RU: 3760288
8 */
9
10 #include <stdlib.h>
11 #include <stdio.h>
12 #include <string.h>
13 #include <ctype.h>
14 #include <locale.h>
15
16 void titulo()
17 {
18     printf("##### Exercício 2 - Atividade Prática #####\n");
19     printf(" Aluno: Paulo Vinicius P. Pinheiro\n");
20     printf(" RU: 3760288\n\n");
21 };
22
23 struct dados
24 {
25     char nome[30];
26     char endereco[50];
27     char telefone[15];
28 };
29
30 int main()
31 {
32     setlocale(LC_ALL, ""); // permite utilizar caracteres em pt-br
33
34     int cont_car = 0, cont_dig = 0, cont = 0;
```

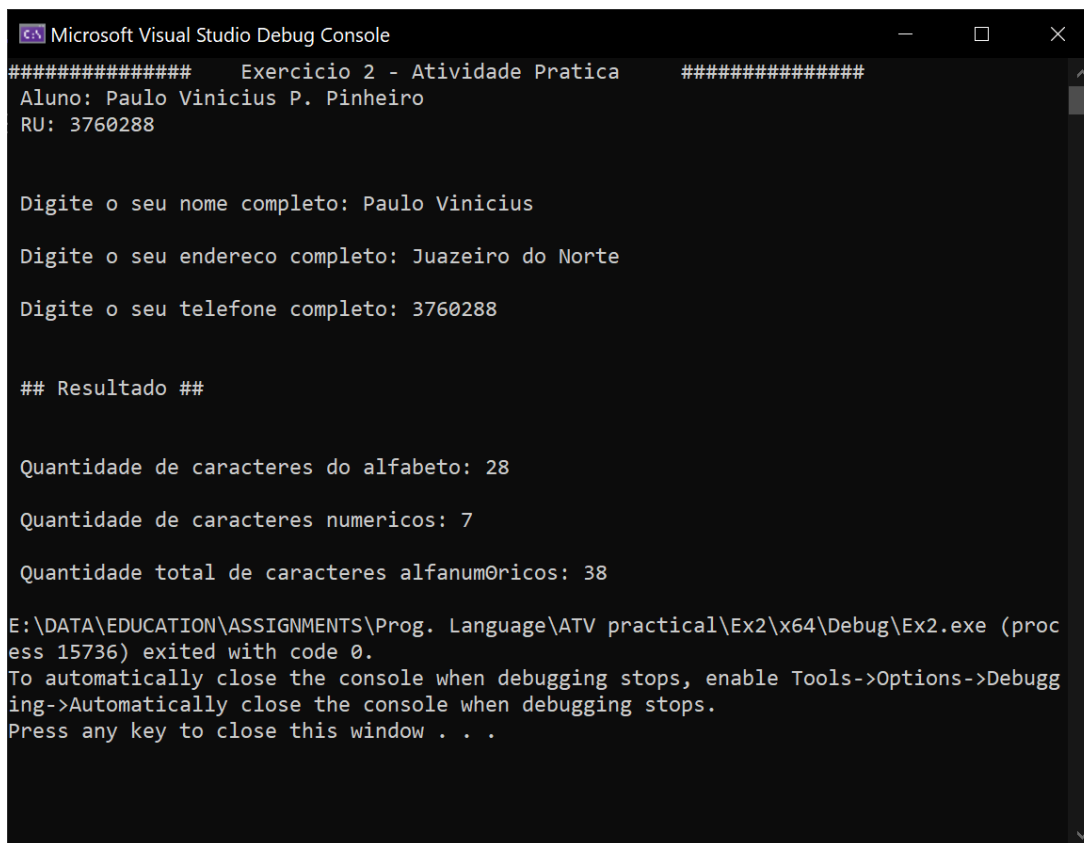
```
35 struct dados pessoa {};  
36 const int tam_total = sizeof(pessoa.nome) + sizeof(pessoa.endereco) +  
    sizeof(pessoa.telefone); // verifica tamanho total as strings  
    concatenadas  
37 char dados_total[tam_total]{};  
38  
39 titulo();  
40  
41 printf("\n Digite o seu nome completo: ");  
42 gets_s(pessoa.nome, 29);  
43  
44 printf("\n Digite o seu endereco completo: ");  
45 gets_s(pessoa.endereco, 49);  
46  
47 printf("\n Digite o seu telefone completo: ");  
48 gets_s(pessoa.telefone, 15);  
49  
50 // como a string final (dados_total) já foi inicializada, apenas  
    realiza-se  
51 //o processo de concatenação dos 3 dados inputados  
52 strcat_s(dados_total, pessoa.nome);  
53 strcat_s(dados_total, pessoa.endereco);  
54 strcat_s(dados_total, pessoa.telefone);  
55  
56 for (int i = 0; i < strlen(dados_total); i++)  
57 {  
58     // ver PVPP  
59     /*if (isalpha(dados_total[i])) printf("\n %c ok\n", dados_total[i]);  
60     else printf("\n %c not ok\n", dados_total[i]);*/  
61  
62     // verifica quais são caracteres alfabéticos e conta-os  
63     if (isalpha(dados_total[i]))  
64     {  
65         cont_car++;  
66         cont++;  
67     }  
68  
69     //verifica quais são caracteres numericos  
70     if (isdigit(dados_total[i]))  
71     {  
72         cont_dig++;  
73         cont++;  
74     }  
75 }  
76  
77 printf("\n\n ## Resultado ##\n\n");  
78 printf("\n Quantidade de caracteres do alfabeto: %d \n", cont_car);  
79 printf("\n Quantidade de caracteres numericos: %d \n", cont_dig);  
80 printf("\n Quantidade total de caracteres digitados: %u \n", strlen(  
    dados_total)-1);  
81  
82 return 0;  
83 }
```

O código fonte apresenta todos os itens solicitados. Dentro do laço for da linha 56, uma verificação é realizada por toda *string* já concatenada (linhas 52-54). As funções *isalpha* e *isdigit* são utilizadas para verificar se o caractere está presente no alfabeto e no dígito, correspondentemente nas linhas 63 e 70. Além do solicitado, também foi desenvolvida a rotina para apresentação da soma dos caracteres alfanuméricos.

2.3 DISCUSSÃO DOS RESULTADOS

Na figura 5, verifica-se a tela de execução do programa com as informações de input e resultados já mostrados.

Figura 5 – Tela final com a segregação dos dados por mês.



```
Microsoft Visual Studio Debug Console
##### Exercício 2 - Atividade Pratica #####
Aluno: Paulo Vinicius P. Pinheiro
RU: 3760288

Digite o seu nome completo: Paulo Vinicius
Digite o seu endereço completo: Juazeiro do Norte
Digite o seu telefone completo: 3760288

## Resultado ##

Quantidade de caracteres do alfabeto: 28
Quantidade de caracteres numericos: 7
Quantidade total de caracteres alfanuméricos: 38

E:\DATA\EDUCATION\ASSIGNMENTS\Prog. Language\ATV practical\Ex2\x64\Debug\Ex2.exe (process 15736) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Conforme apresentado no código fonte e solicitado no enunciado, os caracteres numéricos e alfabéticos são contados, da mesma forma como todos os caracteres alfanuméricos e símbolos válidos.

3 PRÁTICA

3.1 ENUNCIADO

Faça um programa C para calcular o número de lâmpadas 60 watts necessárias para um determinado cômodo. O programa deverá ler um conjunto de informações, tais como: tipo, largura e comprimento do cômodo. O programa termina quando o tipo de cômodo for igual -1. A tabela abaixo mostra, para cada tipo de cômodo, a quantidade de watts por metro quadrado.

Tabela 1 – Dados do problema.

TIPO CÔMODO	POTÊNCIA(W/m ²)
0	12
1	15
2	18
3	20
4	22

3.2 CÓDIGO-FONTE DA SOLUÇÃO

O código fonte desenvolvido como solução para o problema do enunciado está abaixo:

Listing 3.1 – Código fonte da prática 3 - adequação do número de lâmpadas a comodos.

```

1 /*
2 UNINTER - ENGENHARIA DA COMPUTAÇÃO
3 Disciplina de Linguagem de Programação
4 Exercício 3 - Atividade prática
5
6 Autor: Paulo Vinicius Pereira Pinheiro
7 RU: 3760288
8 */
9
10 #include <stdlib.h>
11 #include <stdio.h>
12 #include <string.h>
13 #include <ctype.h>
14 #include <math.h>
15 #include <locale.h>
16
17 void CalulaArea(float* comprimento, float* largura, float* area); // Função
    para calcular a área
18 float Lampada(int tipo, float area); // Função para calcular a qtde de
    lâmpadas de acordo com a área do cômodo
19
20 void titulo()

```

```
21 {
22     printf("##### Exercício 3 - Atividade Prática
23           #####\n");
24     printf(" Aluno: Paulo Vinicius P. Pinheiro\n");
25     printf(" RU: 3760288\n\n");
26     printf("=== Programa para cálculo do número de lâmpadas ===");
27 };
28 // struct para armazenar os dados do cômodo
29 struct dados
30 {
31     int tipo_de_comodo;
32     float comprimento;
33     float largura;
34 };
35 };
36
37 int main()
38 {
39     setlocale(LC_ALL, ""); // Para exibir o caractere 'pt-br' corretamente
40
41     int l = 0;
42     char c = 0;
43     float area;
44     struct dados comodo{};
45
46     titulo();
47
48     while (true) // sempre se repete até atingir a condição de saída: o tipo
49         do comodo deve ser igual a -1
50     {
51         printf("\n\nDigite o tipo do cômodo [0 - 4] ou -1 para sair: "); // -1
52         para sair
53         scanf_s("%d", &comodo.tipo_de_comodo);
54         c = getchar();
55
56         if (comodo.tipo_de_comodo == -1) break;
57
58         printf("\n\nDigite o comprimento do cômodo (em m^2): ");
59         scanf_s("%f", &comodo.comprimento);
60         c = getchar();
61
62         printf("\n\nDigite a largura do cômodo (em m^2): ");
63         scanf_s("%f", &comodo.largura);
64         c = getchar();
65
66         CalulaArea(&comodo.comprimento, &comodo.largura, &area);
67
68         printf("\n\n0 cômodo tem %.2f m^2 e deve possuir %.0f lâmpadas.", area,
69             Lampada(comodo.tipo_de_comodo, area));
70         printf("\n
71             -----\n
72             \n");
73     }
74     printf("\n\nObrigado por utilizar este programa.\n\n");
75 }
76
77 void CalulaArea(float* comprimento, float* largura, float* area)
78 {
79     *area = *comprimento * *largura;
80 }
81
82 float Lampada(int tipo, float area)
```

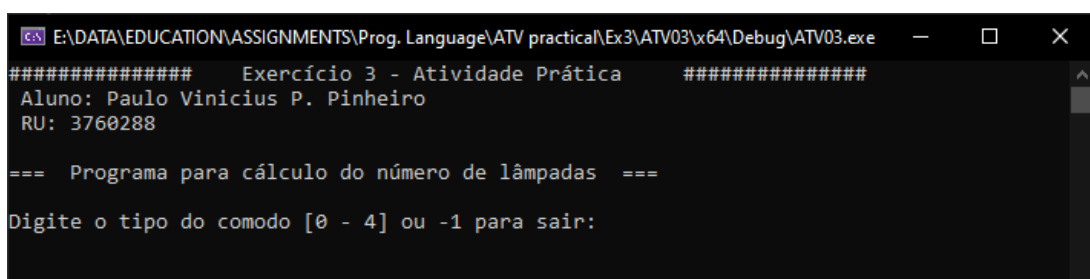
```
78 {
79     float numero_de_lampadas = 0;
80
81     switch (tipo)
82     {
83         case 0:
84             numero_de_lampadas = ceilf(area / 12); //ceilf para o visual studio
85             return numero_de_lampadas;
86         case 1:
87             numero_de_lampadas = ceilf(area / 15);
88             return numero_de_lampadas;
89         case 2:
90             numero_de_lampadas = ceilf(area / 18);
91             return numero_de_lampadas;
92         case 3:
93             numero_de_lampadas = ceilf(area / 20);
94             return numero_de_lampadas;
95         case 4:
96             numero_de_lampadas = ceilf(area / 22);
97             return numero_de_lampadas;
98     default:
99         break;
100 }
101 return 0;
102 }
```

O código fonte apresenta todos os itens solicitados. Dentro do laço *while* da linha 48, todos os inputs são solicitados e o laço se repete até o usuário digitar -1. A área do cômodo e a quantidade de lâmpadas são calculados em funções separadas da main, a partir da linha 72.

3.3 DISCUSSÃO DOS RESULTADOS

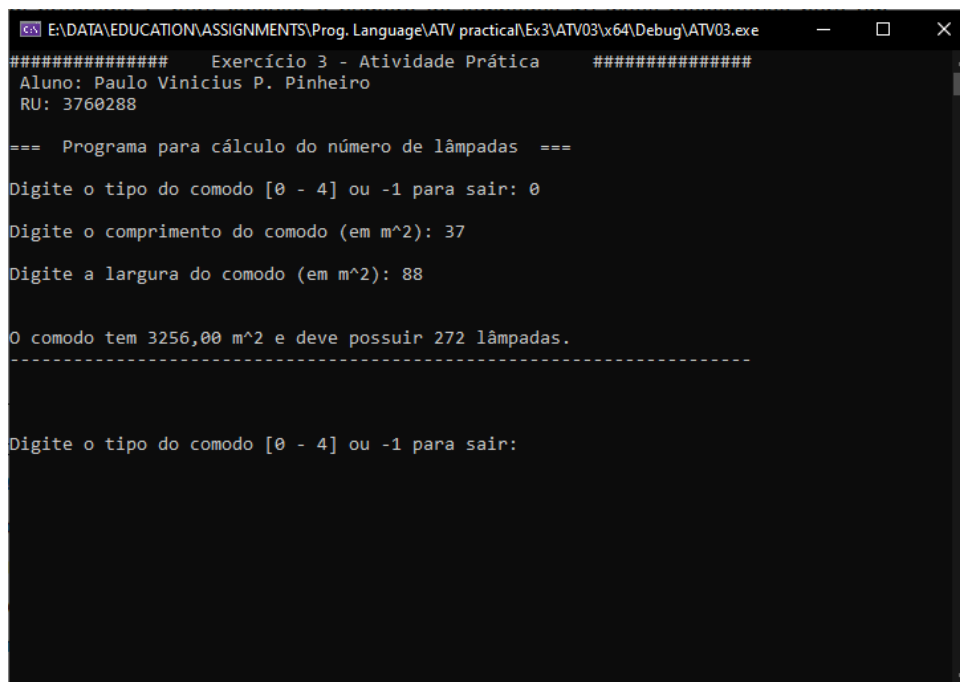
Na figura 6, verifica-se a tela de execução do programa com as informações de input e resultados já mostrados.

Figura 6 – Tela inicial com a solicitação do tipo do cômodo.



A tela inicial, figura 6, conta com um título e apresenta as informações de input solicitadas. A figura XX apresenta a tela com o input dos dados solicitados e o resultado: a quantidade de lâmpadas necessárias para um espaço daquele tipo e com aquelas dimensões.

Figura 7 – Tela final com a apresentação dos resultados.



```
E:\DATA\EDUCATION\ASSIGNMENTS\Prog. Language\ATV practical\Ex3\ATV03\x64\Debug\ATV03.exe
##### Exercício 3 - Atividade Prática #####
Aluno: Paulo Vinicius P. Pinheiro
RU: 3760288

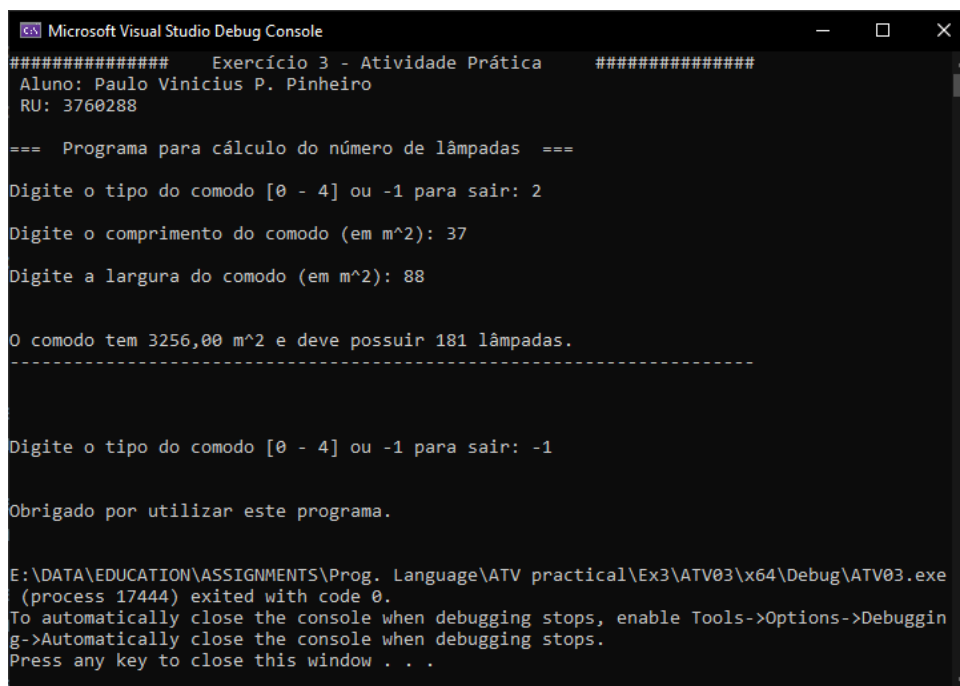
=== Programa para cálculo do número de lâmpadas ===
Digite o tipo do comodo [0 - 4] ou -1 para sair: 0
Digite o comprimento do comodo (em m^2): 37
Digite a largura do comodo (em m^2): 88

O comodo tem 3256,00 m^2 e deve possuir 272 lâmpadas.
-----

Digite o tipo do comodo [0 - 4] ou -1 para sair:
```

Conforme apresentado no código fonte e solicitado no enunciado, o programa calcula a área do cômodo e a quantidade de lâmpadas necessárias para o cômodo escolhido e deve permanecer solicitando esses inputs continuamente até o usuário digitar -1. Na figura 8 está apresentada a tela confirmando a condição de saída do programa.

Figura 8 – Tela final com a apresentação dos resultados.



```
Microsoft Visual Studio Debug Console
##### Exercício 3 - Atividade Prática #####
Aluno: Paulo Vinicius P. Pinheiro
RU: 3760288

=== Programa para cálculo do número de lâmpadas ===
Digite o tipo do comodo [0 - 4] ou -1 para sair: 2
Digite o comprimento do comodo (em m^2): 37
Digite a largura do comodo (em m^2): 88

O comodo tem 3256,00 m^2 e deve possuir 181 lâmpadas.
-----

Digite o tipo do comodo [0 - 4] ou -1 para sair: -1

Obrigado por utilizar este programa.

E:\DATA\EDUCATION\ASSIGNMENTS\Prog. Language\ATV practical\Ex3\ATV03\x64\Debug\ATV03.exe
(process 17444) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```


4 PRÁTICA

4.1 ENUNCIADO

Escreva em linguagem C um algoritmo que:

- Solicite ao usuário que digite o seu RU;
- Armazena cada dígito do Ru em uma posição de um vetor;

Vetor RU	Primeiro dígito						Último dígito
Posição do vetor RU	0	1	2	3	4	5	6

- Utilizando uma função recursiva mostre o qual o valor do menor dígito inserido no vetor.
- Utilizando outra função recursiva mostre o qual o valor do maior dígito inserido no vetor

4.2 CÓDIGO-FONTE DA SOLUÇÃO

O código fonte desenvolvido como solução para o problema do enunciado está abaixo:

Listing 4.1 – Código fonte da prática 4 - verificação do menor e maior dígitos.

```

1 /*
2 UNINTER - ENGENHARIA DA COMPUTAÇÃO
3 Disciplina de Linguagem de Programação
4 Exercício 4 - Atividade prática
5
6 Autor : Paulo Vinicius Pereira Pinheiro
7 RU : 3760288
8
9 */
10
11 #include <stdlib.h>
12 #include <stdio.h>
13 #include <string.h>
14 #include <ctype.h>
15 #include <math.h>
16 #include <clocale>
17
18 int maior(int vet[], int tam, int num_maior);
19 int menor(int vett[], int tamm, int num_menor);

```

```
20
21 void titulo()
22 {
23     printf("##### Exercício 4 - Atividade Prática\n");
24     printf(" Aluno: Paulo Vinicius P. Pinheiro\n");
25     printf(" RU: 3760288\n\n");
26     printf("=== Programa para verificar maior e menor dígito de um RU ===\n");
27 };
28
29 int main()
30 {
31     setlocale(LC_ALL, ""); // para usar acentos
32
33     int cont = 0, n_maior = 0, n_menor = 0;
34     int ru[7]{};
35     char c;
36
37     titulo();
38     printf("\n\nDigite seu RU dígito a dígito.\n\n ");
39
40     while (cont <= 6) //laço para input dos dados
41     {
42         printf("\nDigite o dígito número %i: ", cont + 1);
43         scanf_s("%d", &ru[cont]);
44         c = getchar();
45         cont++;
46     }
47     n_menor = menor(ru, 7, 9);
48     printf("\n0 menor dígito do RU digitado é %d.\n\n", n_menor);
49
50     n_maior = maior(ru, 7, 0);
51     printf("\n0 maior dígito do RU digitado é %d.\n\n", n_maior);
52
53     return 0;
54 }
55
56 //função recursiva para verificar o maior dígito
57 int maior(int vet[], int tam, int num_maior)
58 {
59     if (tam == 0)
60     {
61         return num_maior;
62     }
63     else
64     {
65         if (vet[tam] > num_maior)
66         {
67             num_maior = vet[tam];
68             return maior(vet, tam - 1, num_maior);
69         }
70         else
71         {
72             return maior(vet, tam - 1, num_maior);
73         }
74     }
75 }
76 // função recursiva para verificar o menor dígito
77 int menor(int vett[], int tamm, int num_menor)
78 {
79     if (tamm == 0)
80     {
```

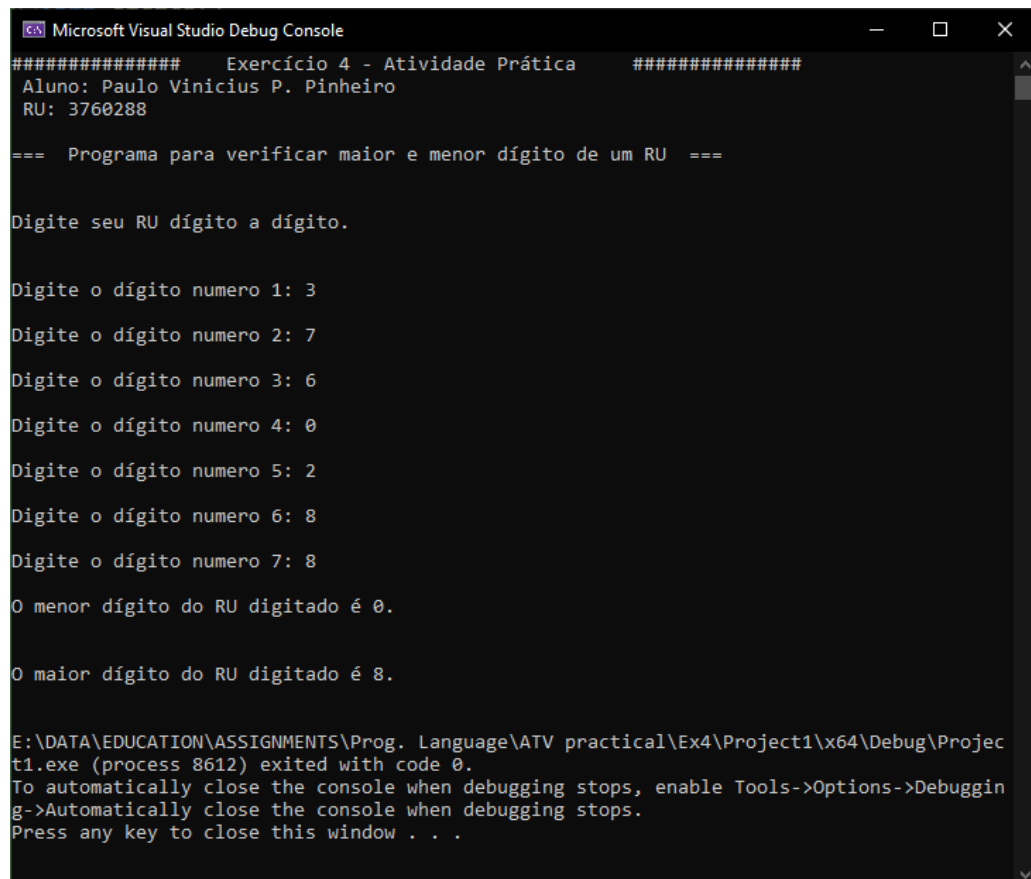
```
81     return num_menor;
82 }
83 else
84 {
85     if (vett[tamm-1] < num_menor)
86     {
87         num_menor = vett[tamm-1];
88         return menor(vett, tamm - 1, num_menor);
89     }
90     else
91     {
92         return menor(vett, tamm - 1, num_menor);
93     }
94 }
95 }
```

O código fonte apresenta todos os itens solicitados. Após a função principal, estão inseridas as funções *menor* e *maior* para verificar o menor e o maior dígito do vetor RU.

4.3 DISCUSSÃO DOS RESULTADOS

Na figura 9, verifica-se a tela de execução do programa com as informações de input e resultados já mostrados.

Figura 9 – Tela com a solicitação do vetor de entrada e resultados.



```
Microsoft Visual Studio Debug Console
##### Exercício 4 - Atividade Prática #####
Aluno: Paulo Vinicius P. Pinheiro
RU: 3760288

=== Programa para verificar maior e menor dígito de um RU ===

Digite seu RU dígito a dígito.

Digite o dígito numero 1: 3
Digite o dígito numero 2: 7
Digite o dígito numero 3: 6
Digite o dígito numero 4: 0
Digite o dígito numero 5: 2
Digite o dígito numero 6: 8
Digite o dígito numero 7: 8

O menor dígito do RU digitado é 0.

O maior dígito do RU digitado é 8.

E:\DATA\EDUCATION\ASSIGNMENTS\Prog. Language\ATV practical\Ex4\Project1\x64\Debug\Projec
t1.exe (process 8612) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debuggin
g->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

5 PRÁTICA

5.1 ENUNCIADO

Crie um programa, em linguagem C, que receba 7 registros contendo, Nome do Produto, Código do produto (numérico), valor do produto. 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 de campo. O nome do arquivo deve ser o seu número de RU.

5.2 CÓDIGO-FONTE DA SOLUÇÃO

O código fonte desenvolvido como solução para o problema do enunciado está abaixo:

Listing 5.1 – Código fonte da prática 5 - armazena dados em arquivo .csv.

```

1 /*
2 UNINTER - ENGENHARIA DA COMPUTAÇÃO
3 Disciplina de Linguagem de Programação
4 Exercício 5 - Atividade prática
5
6 Autor : Paulo Vinicius Pereira Pinheiro
7 RU : 3760288
8
9 */
10
11 #include <stdlib.h>
12 #include <stdio.h>
13 #include <string.h>
14 #include <ctype.h>
15 #include <math.h>
16 #include <locale.h>
17
18 FILE* stream; //stream de entrada para arquivo gerado pelo usuário
19
20 struct dados
21 {
22     char nome[50]{};
23     int codigo;
24     float preco;
25 };
26
27 void titulo()
28 {
29     printf("##### Exercício 5 - Atividade Prática\n");
30     printf(" Aluno: Paulo Vinicius P. Pinheiro\n");
31     printf(" RU: 3760288\n\n");
32     printf("=== Programa para armazenar dados em arquivos CSV ===\n");
33 };
34
35 int main()

```

```
36 {
37     setlocale(LC_ALL, ""); // permite usar a lingua atual de escrita
38
39     errno_t err;
40     char nome_arquivo[] = "3760288.csv";
41     char c;
42     const int num_prod = 7;
43     struct dados produto[num_prod]{};
44
45     // cria o arquivo .csv com o número do RU
46     err = fopen_s(&stream, nome_arquivo, "w");
47     if (err == 0)
48     {
49         printf("O arquivo criado com sucesso.\n");
50         fprintf(stream, "código_produto; nome_produto; preco_produto\n");
51     }
52     else
53     {
54         printf("Erro ao criar o arquivo.\n");
55         exit(1);
56     }
57
58     titulo();
59     for (int i = 0; i < num_prod; i++)
60     {
61         printf("\n==== Registro número %d =====\n", i + 1);
62         printf("\nDigite o código do produto: ");
63         scanf_s("%d", &produto[i].codigo);
64         while ((c = getchar()) != '\n' && c != EOF){}
65         // trecho para evitar sobresalto ao digitar os valores
66
67         printf("\nDigite o nome do produto: ");
68         gets_s(produto[i].nome, 50);
69         c = getchar();
70
71         printf("\nDigite o preço do produto: ");
72         scanf_s("%f", &produto[i].preco);
73         while ((c = getchar()) != '\n' && c != EOF) {}
74
75         // escreve no arquivo os dados de input
76         fprintf(stream, "%d; %s; %.2f\n", produto[i].codigo, produto[i].nome,
77             produto[i].preco);
78     }
79     // fecha o arquivo criado
80     _fcloseall();
81     return 0;
82 }
```

O código fonte apresenta todos os itens solicitados. O nome do arquivo .csv para armazenamento dos dados é previamente definido pelo desenvolvedor (linha 40). Porém, o mesmo poderia ser solicitado ao usuário. Após a linha 45 é realizada a criação do arquivo com verificação de erro. No laço *for* iniciado na linha 59, é iniciada a solicitação de entrada dos dados e consequente inscrição no arquivo .csv. Na linha 76, têm se a instrução de escrita no arquivo.

5.3 DISCUSSÃO DOS RESULTADOS

Na figura 10, verifica-se a tela de execução do programa com os registros de entrada devidamente separados.

Figura 10 – Tela com a solicitação dos registros de entrada.

```

E:\DATA\EDUCATION\ASSIGNMENTS\Prog. Language\ATV practical\Ex5\Project1\Debug\Project1.exe
O arquivo criado com sucesso.
##### Exercício 5 - Atividade Prática #####
Aluno: Paulo Vinicius P. Pinheiro
RU: 3760288

=== Programa para armazenar dados em arquivos CSV ===

==== Registro número 1 =====
Digite o código do produto: 01
Digite o nome do produto: produto 01
Digite o preço do produto: 12.35

==== Registro número 2 =====
Digite o código do produto: 02
Digite o nome do produto: produto 02
Digite o preço do produto: 45.32

```

Na figura 11, verifica-se a criação do arquivo com o RU 3760288.csv e a correta inscrição dos dados de input separados por ponto e vírgula.

Figura 11 – Tela com a criação do arquivo .csv e a inscrição dos dados de input.

Name	Date modified	Type	Size
.vs	02/08/2022 21:34	File folder	
x64	02/08/2022 22:23	File folder	
3760288.csv	02/08/2022 23:17	Microsoft Excel C...	1 KB
Project1.sln	02/08/2022 21:34	Visual Studio Solu...	2 KB
Project1.vcxproj	02/08/2022 21:51	VC++ Project	7 KB
Project1.vcxproj.filters	02/08/2022 21:51	VC++ Project Filte...	1 KB
Project1.vcxproj.user	02/08/2022 21:34	Per-User Project O...	1 KB
Source.cpp	02/08/2022 23:12	C++ Source File	2 KB

```

3760288.csv
1 código_produto; nome_produto; preco_produto
2 1; produto 01; 12,00
3 2; produto 02; 45,00
4 3; produto 03; 34,00
5 4; produto 04; 20,00
6 5; produto 05; 34,00
7 6; produto 06; 65,00
8 7; produto 07; 9,00
9

```

CONCLUSÃO

Através da confecção desta atividade prática, o aluno teve a oportunidade de aprimorar a utilização da linguagem C e todos os conceitos apresentados na disciplina de Linguagem de programação. Estruturas de repetição, estruturas de dados e funções recursivas são alguns dos pontos abordados nestes cinco exercícios práticos.

Todos os exercícios foram desenvolvidos com a linguagem C no IDE de desenvolvimento Microsoft Visual Studio 2022(Community).

Este documento e seu código-fonte foram produzidos em L^AT_EX e são exemplos de referência de uso da classe `abntex2` e do pacote `abntex2cite`. O documento exemplifica a elaboração de trabalho acadêmico (tese, dissertação e outros do gênero) produzido conforme a ABNT NBR 14724:2011 *Informação e documentação - Trabalhos acadêmicos - Apresentação*.