

Coleção Introdução à Linguagem C

Volume 2

Linguagem C

Aprendendo com Exercícios Resolvidos

Autores

André Barros de Sales

Arthur de Moura Del Esposte

Organizadora

Márcia Barros de Sales

Coleção Introdução à Linguagem C

Volume 2

Linguagem C

Aprendendo com Exercícios Resolvidos

Coleção Introdução à Linguagem C

Volume 2

Linguagem C

Aprendendo com Exercícios Resolvidos

Autores

André Barros de Sales

Arthur de Moura Del Esposte

Organizadora

Márcia Barros de Sales



Florianópolis
2017

2017 Dos autores



Esta obra é disponibilizada nos termos da Licença Creative Commons Atribuição–NãoComercial–SemDerivações 4.0 Internacional. É permitida a reprodução parcial ou total desta obra, desde que citada a fonte.

Coordenação Gráfica

Denise Aparecida Bunn

Capa, Projeto Gráfico e Diagramação

Claudio José Girardi

Revisão Textual

Márcia Regina Pereira Sagaz

Ficha Catalográfica

S1631 Sales, André Barros de

Linguagem C [Recurso eletrônico on-line] : aprendendo com exercícios resolvidos / André Barros de Sales, Arthur de Moura Del Esposte; organizadora Márcia Barros de Sales. – Florianópolis : Departamento de Ciências da Administração / UFSC, 2017.

224 p. : il. , figs. – (Introdução à Linguagem C; v.2)

ISBN: 978-85-7988-329-3

ISBN COLEÇÃO COMPLETA: 978-85-7988-310-1

Inclui bibliografia

Modo de acesso: <http://repositorio.unb.br/handle/10482/25306>.

1. C (Linguagem de programação de computadores) – Estudo e ensino.
 2. C (Linguagem de programação de computadores) – Exercícios.
- I. Del Esposte, Arthur de Moura. II. Sales, Márcia Barros de. III. Título.
IV. Série.

CDU: 681.31.06C

Prefácio

A “Coleção Introdução à Linguagem C” segue sua jornada, e os autores André Barros de Sales e Arthur de Moura Del Esposte agora propõem a obra “Aprendendo com Exercícios Resolvidos”, visando auxiliar os discentes de Computação, Engenharia e de áreas afins a programar em linguagem C por meio de exercícios resolvidos de programação.

Atualmente, existem meios e recursos didáticos acessíveis muito bons para o aprendizado da linguagem C fora da sala de aula como: tutoriais em canais do YouTube, blogs específicos e fóruns de discussão; grupos de Pesquisa & Desenvolvimento (P&D); e livro de exercícios resolvidos, em formato impresso ou e-book.

Então surge a pergunta: por que ensinar com exercícios resolvidos? Porque os autores, ao longo de sua prática docente, observaram que alguns de seus estudantes aprendiam a programar em C com muito mais facilidade e efetividade por meio de exercícios resolvidos. Assim, ensinar linguagem C de forma prática, objetiva e intuitiva tornou-se um desafio para esses autores, que já registram resultados positivos, atraindo novos interessados nessa matéria.

Nesta obra, reitero, os autores, fazendo uso de um método educativo agradável, interessante e intuitivo, convidam o discente a conhecer uma seleção primorosa de exercícios resolvidos, suprindo, assim, a deficiência daquele que deseja colocar em prática a teoria adquirida.

Georges Daniel Amvame Nze

Professor Adjunto

Coordenador do curso de Graduação de Engenharia de Redes de Comunicação

Departamento de Engenharia Elétrica (EnE)

Faculdade de Tecnologia (FT)

Universidade de Brasília (UnB)

Sumário

Apresentação.....	1
Capítulo 1	
Estrutura Sequencial – Comandos Básicos	
Comandos Básicos: exercícios	5
Capítulo 2	
Estrutura Sequencial – Tipos de Dados e Variáveis	
Tipos de Dados e Variáveis: exercícios.....	13
Capítulo 3	
Estrutura Sequencial – Variáveis e Operadores	
Variáveis e Operadores: exercícios.....	19
Capítulo 4	
Estrutura de Controle de Fluxo – Seleção Simples	
Seleção Simples: exercícios	27
Capítulo 5	
Estrutura de Controle de Fluxo – Condicional Composta	
Condicional Composta: exercícios	49

Capítulo 6**Estrutura de Repetição Contada**

Estrutura de Repetição Contada: exercícios 79

Capítulo 7**Condicional com Teste no Início**

Condicional com Teste no Início: exercícios 93

Capítulo 8**Estrutura de Repetição – Condicional com Teste no Final**

Condicional com Teste no Final: exercícios 111

Capítulo 9**Funções**

Funções: exercícios 123

Capítulo 10**Vetores**

Vetores: exercícios 157

Capítulo 11**Matrizes**

Matrizes: exercícios 171

Capítulo 12**Strings**

Strings: exercícios 187

Capítulo 13**Ponteiros**

Ponteiros: exercícios 209

Bibliografia 222

Apresentação

O livro “Linguagem C – Aprendendo com Exercícios Resolvidos” é o segundo volume da “Coleção Introdução à Linguagem C”, que foi concebida para oferecer à comunidade acadêmica e aos demais interessados, de forma clara, objetiva e explicativa, exemplos de exercícios resolvidos em linguagem C.

São mais de 100 exercícios estruturados em 13 capítulos que respeitam a sequência de apresentação dos conteúdos do primeiro volume da coleção. Além dos exercícios do primeiro volume da coleção, há novos exercícios e suas respectivas soluções, todos ordenados por diferentes níveis de dificuldade, possibilitando ao discente um desenvolvimento gradual e consistente. Propositalmente, alguns exercícios são repetidos em diferentes capítulos, porém com demandas específicas referentes ao conteúdo do cada capítulo. Recomenda-se rever os conteúdos de programação abordados no livro “Linguagem C – Roteiros de Experimentos para Aulas Práticas” para tornar este aprendizado mais efetivo.

É importante ressaltar que este livro resulta de uma pesquisa bem-sucedida da utilização técnica de Desing Centrado no Usuário (DCU) com discentes da disciplina de programa-

ção de “Linguagem C” dos cursos de Engenharias da Faculdade UnB Gama – FGA da Universidade de Brasília (UnB). Dividida em duas fases, essa pesquisa contou com a participação de 90 discentes do segundo semestre de 2010 e com mais 108 discentes do primeiro semestre de 2011.

Baseado nessa técnica de DCU, os discentes avaliaram, nas primeiras versões desta obra, a formatação (tamanho das figuras, linguagem, facilidade do manuseio e tamanho das letras e das figuras), o conteúdo (explicações/comentários, resolução dos exercícios e quantidade dos exercícios) e a aprendizagem (se eles seriam capazes de explicar todas ou algumas das soluções dos exercícios apresentadas a outros discentes). No experimento, os discentes da primeira fase do estudo sugeriram ajustes na forma e no conteúdo que foram avaliados e validados pelos discentes da segunda fase e pelos autores.

Certos de que o leitor desfrutará do bom resultado desse experimento em sala de aula, os autores salientam que todo exercício de programação pode ser resolvido de muitas maneiras, ou seja, diferentes das que são apresentadas aqui. Os autores recomendam primeiro a resolução do exercício e, somente depois, a conferência da resposta, tornando mais simples a interpretação dos códigos e eficiente o aprendizado da linguagem C.

Uma ótima leitura a todos!

Márcia Barros de Sales
Organizadora da Coleção

Capítulo 1

**Estrutura Sequencial
Comandos Básicos**

Comandos Básicos

- 1.1 Escreva um programa que leia o número de alunos e de alunas de uma sala. Como saída, o programa deve apresentar o número de alunos e em seguida o de alunas.

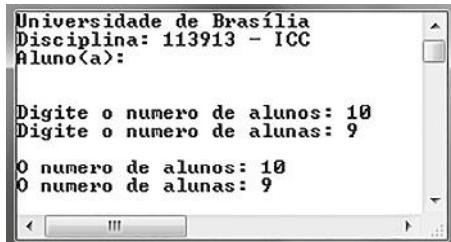
Resposta: o código do comando é apresentado na Figura 1.1 e o resultado da execução desse código na Figura 1.2.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de Variáveis
6     int num_alunos, num_alunas;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Digite o numero de alunos: "); //Início de interação
13    scanf("%d", &num_alunos); /*Receber o dado com a função scanf do
14 tipo int identificado por %d que será armazenado no espaço reservado para
15 num_alunos, &num_alunos*/
16    printf("Digite o numero de alunas: ");// Segunda impressão
17    scanf ("%d", &num_alunas); /* Receber o dado do tipo int identificado por %d
18 que será armazenado no espaço reservado para num_alunos, &num_alunas */
19    printf("\nO numero de alunos: %d", num_alunos); /*Apresentação de resultados
20 na tela com a função printf. Onde está o %d, será apresentado o valor de uma
21 variável de tipo int, que por sua vez é identificada posteriormente, num_alunos,
22 por fim, \n significa que deseja-se pular uma linha.*/
23    printf("O numero de alunas: %d", num_alunas);
24    getch(); //Função que aguarda o recebimento de um caractere via teclado.
25 }
```

Figura 1.1 – Código do programa em C: *respexe11.cpp*

Fonte: Elaboração própria

Figura 1.2 – Resultado da execução do código do programa em C: *respexe11.cpp*

Fonte: Elaboração própria

- 1.2 Execute novamente o programa, de modo que seja apresentado primeiro o número de alunas e depois o número de alunos.

Resposta: a nova sequência de comandos é apresentada na Figura 1.3 e o resultado da execução desse código na Figura 1.4.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de Variáveis
6     int num_alunos, num_alunas;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Digite o numero de alunos: "); //Início de interação
13    scanf("%d", &num_alunos); /*Receber o dado com a função scanf do
14 tipo int identificado por %d que será armazenado no espaço reservado para
15 num_alunos, &num_alunas*/
16    printf("Digite o numero de alunas: ");// Segunda impressão
17    scanf ("%d", &num_alunas); /* Receber o dado do tipo int identificado por %d
18 que será armazenado no espaço reservado para num_alunos, &num_alunas */
19
20    /* Para apresentar os resultados como pedido, basta trocar os comandos de
21 posição. O programa é executado com os comandos sequencialmente, ou seja,
22 este executa primeiramente o que vem antes e posteriormente o que vem depois.*/
23    printf("\nO numero de alunas: %d", num_alunas);
24    printf("O numero de alunos: %d", num_alunos);
25    getch(); //função que aguarda o recebimento de um caractere via teclado.
26 }
```

Figura 1.3 – Código do programa em C: *respexe12.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o numero de alunos: 20
Digite o numero de alunas: 22

O numero de alunas: 22
O numero de alunos: 20
```

Figura 1.4 – Resultado da execução do código do programa em C: *respexe12.cpp*

Fonte: Elaboração própria

- 1.3 Escreva um programa em C que apresente a tabuada do número 3, conforme a Figura 1.5.

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Tabuada de 3:
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```

Figura 1.5 – Resultado da execução do código do programa em C a ser elaborado: *respexe13.cpp*.

Fonte: Elaboração própria

Resposta: o código do programa é apresentado na Figura 1.6.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5
6     //Cabeçalho
7     printf("Universidade de Brasília\n", 161);
8     printf("Disciplina: 113913 - ICC \n");
9     printf("Aluno(a):\n\n");
10    // Início da lógica de programação
11    printf("Tabuada de 3:\n\n");
12    // Escrevendo Mensagens na Tela
13    printf("\t3 x 1 = %d\n", 3*1);
14    printf("\t3 x 2 = %d\n", 3*2);
15    printf("\t3 x 3 = %d\n", 3*3);
16    printf("\t3 x 4 = %d\n", 3*4);
17    printf("\t3 x 5 = %d\n", 3*5);
18    printf("\t3 x 6 = %d\n", 3*6);
19    printf("\t3 x 7 = %d\n", 3*7);
20    printf("\t3 x 8 = %d\n", 3*8);
21    printf("\t3 x 9 = %d\n", 3*9);
22    printf("\t3 x 10 = %d\n", 3*10);
23    printf("\n\n");
24
25 } //Função que aguarda o recebimento de um caractere via teclado.
26

```

Figura 1.6 – Código do programa em C: *respexe13.cpp*

Fonte: Elaboração própria

- 1.4 Escreva um programa em C para o Ministério da Saúde que o auxilie nas informações sobre a dengue em Brasília. Esse programa deve receber os dados sobre o número de casos suspeitos, o número de casos confirmados e o número de mortes, apresentando a tela, conforme Figura 1.7.

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Preencha com as seguintes informacoes sobre a Dengue em Brasilia:
    Casos Suspeitos: 12
    Casos Confirmados: 15
    Quantidade de Mortes: 2

Informacoes sobre a dengue em Brasilia:
    Casos Suspeitos: 12
    Casos Confirmados: 15
    Quantidade de Mortes: 2

    Total de dados: 29
  
```

Figura 1.7 – Resultado da execução do código do programa em C a ser elaborado:

respexe14.cpp

Fonte: Elaboração própria

Resposta: o código do programa é apresentado na Figura 1.8.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de Variáveis
6     int total, susp, mort, conf;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    // Início da lógica de programação
12    printf("Preencha com as seguintes informacoes sobre a Dengue em Brasilia:\n\n");
13    printf("\tCasos Suspeitos: ");
14    scanf("%d", &susp);
15    printf("\tCasos Confirmados: ");
16    scanf("%d", &conf);
17    printf("\tQuantidade de Mortes: ");
18    scanf("%d", &mort);
19    // Apresentação dos resultados
20    printf("\nInformações sobre a dengue em Brasilia:\n");
21    printf("\tCasos Suspeitos: %d\n", susp);
22    printf("\tCasos Confirmados: %d\n", conf);
23    printf("\tQuantidade de Mortes: %d\n", mort);
24    total=susp+conf+mort;
25    printf("\n\tTotal de dados: %d\n", total);
26
27
28 } getch(); //Função que aguarda o recebimento de um caractere via teclado.
  
```

Figura 1.8 – Código do programa em C: *respexe14.cpp*

Fonte: Elaboração própria

Capítulo 2

Estrutura Sequencial
Tipos de Dados e Variáveis

Tipos de Dados e Variáveis

- 2.1 Escreva um programa que leia o número de chuteiras de uma loja de esporte. Os valores deverão ser inseridos por meio do teclado. Como saída, o programa deve apresentar o número de chuteiras e suas marcas (*marca A*, *marca B* e *marca C*).

Utilizando o caráter especial `\t` dentro do *printf*, desejamos que você obtenha uma saída parecida com esta:

Quantidade de chuteiras em estoque

Marca (A)	Marca (B)	Marca (C)
10	25	45

Resposta: o código é apresentado na Figura 2.1 e o resultado da execução desse código na Figura 2.2.

```
1 #include<stdio.h> // Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> // Biblioteca para manipulação de caracteres
3 #include<stdlib.h> // Biblioteca padrão para uso de funções do sistema
4
5 int main ()//Programa principal
6     // Declaração de Variáveis
7     int marcaA, marcaB, marcaC; // Variáveis do tipo inteiro para marca A, B e C.
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12
13 /* Os seguintes comandos printf são para imprimir a mensagem dentro das "" para que haja
14 comunicação entre computador e o usuário do programa. As funções scanf sevem para receber os números desejados, o "%d" indica que a variável ao qual será destinada o valor
15 recebido é do tipo inteiro. o % indica que será armazenado no espaço declarado anterior-
16 mente para a variável que vem logo em seguida deste %. */
17    printf("Digite a quantidade de chuteiras da marca A: ");
18    scanf("%d", &marcaA);
19    printf("\nDigite a quantidade de chuteiras da marca B: ");
20    scanf("%d", &marcaB);
21    printf("\nDigite a quantidade de chuteiras da marca C: ");
22    scanf("%d", &marcaC);
23
24 //Apresentação dos Resultados Letra A)
25    printf("\nExistem %d chuteiras da marca A\n", marcaA);
26    printf("Existem %d chuteiras da marca B\n", marcaB);
27    printf("Existem %d chuteiras da marca C\n", marcaC);
28 //Apresentação dos Resultados Letra B)
29    printf("Existem %d chuteiras da marca A, %d da marca B, %d da marca C\n\n", marcaA,marcaB,marcaC);
30 // Apresentação dos Resultados Letra C)
31    printf("\t\t Quantidade de Chuteiras em Estoque\n");
32    printf("\t\t Marca A \t Marca B \t Marca C\n");
33    printf("\t\t %d \t %d \t %d\n", marcaA,marcaB,marcaC);
34    getch(); //Função que aguarda o recebimento de um caractere via teclado.
```

Figura 2.1 – Código do programa em C: *respexe21.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite a quantidade de chuteiras da marca A: 10
Digite a quantidade de chuteiras da marca B: 20
Digite a quantidade de chuteiras da marca C: 30

Existem 10 chuteiras da marca A
Existem 20 chuteiras da marca B
Existem 30 chuteiras da marca C

Existem 10 chuteiras da marca A, 20 da marca B, 30 da marca C

    Quantidade de Chuteiras em Estoque
    Marca A      Marca B      Marca C
    10          20          30

```

Figura 2.2 – Resultado da execução do código do programa em C: *respexe21.cpp*

Fonte: Elaboração própria

2.2 Escreva um programa em C que apresente a tabuada de um número qualquer, apresentando a tela, conforme a Figura 2.3, sendo que esse programa deverá apresentar o seguinte trecho de código para apresentação dos dados:

```

printf("Tabuada de %d\n\n", num);
// Escrevendo Mensagens na Tela
printf("\t%d x 1 = %d\n", num, num*1);
printf("\t%d x 2 = %d\n", num, num*2);
printf("\t%d x 3 = %d\n", num, num*3);

```

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite um numero inteiro: 9
Tabuada de 9:

    9 x 1 = 9
    9 x 2 = 18
    9 x 3 = 27
    9 x 4 = 36
    9 x 5 = 45
    9 x 6 = 54
    9 x 7 = 63
    9 x 8 = 72
    9 x 9 = 81
    9 x 10 = 90

```

Figura 2.3 – Resultado da execução do código do programa em C a ser elaborado:

respexe22.cpp

Fonte: Elaboração própria

Resposta: o código do programa é apresentado na Figura 2.4.

```
1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de Variáveis
6     int num;
7     //Cabecalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    // Início da lógica de programação
12    printf("Digite um numero inteiro: ");
13    scanf("%d", &num);
14    printf("Tabuada de %d:\n\n", num);
15    // Escrevendo Mensagens na Tela
16    printf("\t%d x 1 = %d\n", num, num*1);
17    printf("\t%d x 2 = %d\n", num, num*2);
18    printf("\t%d x 3 = %d\n", num, num*3);
19    printf("\t%d x 4 = %d\n", num, num*4);
20    printf("\t%d x 5 = %d\n", num, num*5);
21    printf("\t%d x 6 = %d\n", num, num*6);
22    printf("\t%d x 7 = %d\n", num, num*7);
23    printf("\t%d x 8 = %d\n", num, num*8);
24    printf("\t%d x 9 = %d\n", num, num*9);
25    printf("\t%d x 10 = %d\n", num, num*10);
26    getch(); //Função que aguarda o recebimento de um caractere via teclado.
27 }
```

Figura 2.4 – Código do programa em C: *respexe22.cpp*

Fonte: Elaboração própria

Capítulo 3

**Estrutura Sequencial
Variáveis e Operadores**

Variáveis e Operadores

3.1 Escreva um programa que leia um número na base decimal e em seguida imprima esse mesmo número nas bases octal e hexadecimal.

Resposta: o código do programa é apresentado na Figura 3.1. e o resultado da execução desse código é apresentado na Figura 3.2 e na Figura 3.3.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de Variáveis
6     int num; //variável que irá receber o número na base decimal
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Escrevendo o comando na tela
12    printf("Escreva um numero na base decimal: ");
13    //recebendo do teclado o número decimal
14    scanf("%d", &num);
15
16    //Imprimindo o número lido nas bases: octal e hexadecimal
17    printf("\nO numero %d na base octal e: %o\n", num, num);
18    printf("O numero %d na base hexadecimal e: %x", num, num);
19    getch(); //Função que aguarda o recebimento de um caractere via teclado.
20 }
```

Figura 3.1 – Código do programa em C: *respexe31.cpp*

Fonte: Elaboração própria

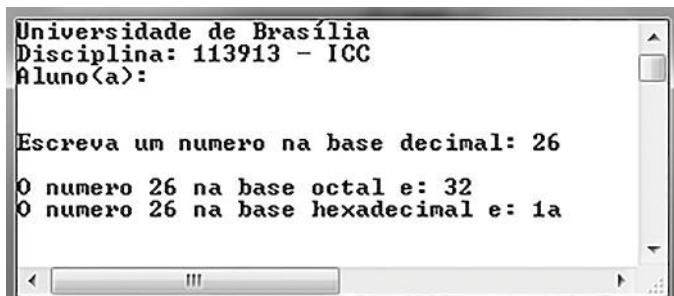


Figura 3.2 – Resultado da execução do código do programa em C: *respexe31.cpp*, que ilustra um exemplo em que o número digitado é 26

Fonte: Elaboração própria

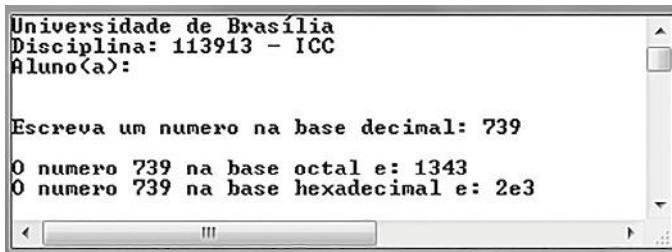


Figura 3.3 – Resultado da execução do código do programa em C: *respexe31.cpp*, que ilustra um outro exemplo em que o número digitado é 739

Fonte: Elaboração própria

3.2 Desenvolva um programa para calcular e escrever a área e o perímetro de um quadrado. Dados de entrada: tamanho do lado (tipo da variável: numérico inteiro, valor em centímetros). Essa conversão deve ser feita por meio da função printf().

Resposta: o código do programa é apresentado na Figura 3.4 e o resultado da execução desse código na Figura 3.5.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main () //Programa principal
5     // Declaração de Variáveis
6     int lado, area, perimetro;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    // Início da lógica de programação
12    printf("Digite o lado do quadrado em cm: ");
13    scanf("%d", &lado);
14    //Cálculos
15    area=lado*lado;
16    perimetro=lado*4;
17    printf("\nA área do quadrado é: %d", area);
18    printf("\nO perímetro do quadrado é: %d", perimetro);
19    getch(); //Função que aguarda o recebimento de um caractere via teclado.
20 }
```

Figura 3.4 – Código do programa em C: *respexe32.cpp*

Fonte: Elaboração própria

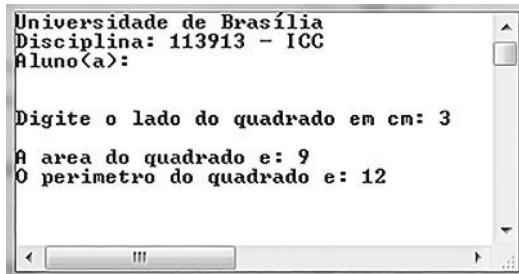


Figura 3.5 – Resultado da execução do programa em C: *respexe32.cpp*

Fonte: Elaboração própria

3.3 Desenvolva um programa para calcular e escrever a área e o perímetro de um retângulo. Dados de entrada: tamanho da base e da altura (tipo das variáveis: inteiro, valor em centímetros).

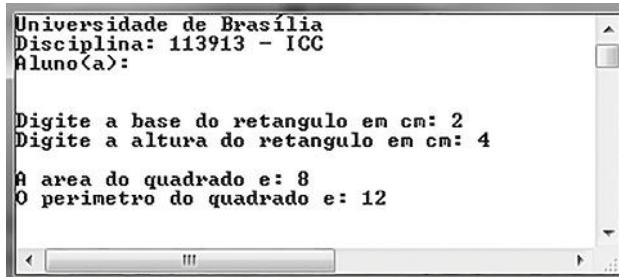
Resposta: o código do programa é apresentado na Figura 3.6 e o resultado da execução desse código na Figura 3.7.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h>/Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de Variáveis
6     int base,altura, area, perimetro;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    // Início da lógica de programação
12    printf("Digite a base do retângulo em cm: ");
13    scanf("%d", &base);
14    printf("Digite a altura do retângulo em cm: ");
15    scanf("%d", &altura);
16    //Cálculos
17    area=base*altura;
18    perimetro=(base*2)+(altura*2);
19    printf("\nA área do quadrado e: %d", area);
20    printf("\nO perímetro do quadrado e: %d", perimetro);
21    getch(); //Função que aguarda o recebimento de um caractere via teclado.
22 }
```

Figura 3.6 – Código do programa em C: *respexe33.cpp*

Fonte: Elaboração própria

Figura 3.7 – Resultado da execução do código do programa em C: *respexe33.cpp*

Fonte: Elaboração própria

- 3.4 Escreva um programa que solicite ao usuário dois números inteiros diferentes de zero. Com esses números imprima na tela: a soma, a subtração, o produto, a divisão e a média.

Resposta: o código do programa é apresentado na Figura 3.8 e o resultado da execução desse código na Figura 3.9.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main () //Programa principal
5     // Declaração de Variáveis
6     int num1, num2;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    // Início da lógica de programação
12    printf("Você deverá digitar dois números diferentes de zero\n\n");
13    printf("Digite o primeiro número: ");
14    scanf("%d", &num1);
15    printf("Digite o segundo número: ");
16    scanf("%d", &num2);
17    printf("\nA soma %d + %d = %d", num1, num2, num1+num2);
18    printf("\nA subtração %d - %d = %d", num1, num2, num1-num2);
19    printf("\nO produto %d x %d = %d", num1, num2, num1*num2);
20    printf("\nA divisão %d : %d = %d", num1, num2, num1/num2);
21    printf("\nA média entre %d e %d = %.2f", num1, num2, (float) (num1+num2)/2);
22    getch(); //Função que aguarda o recebimento de um caractere via teclado
23 }
```

Figura 3.8 – Código do programa em C: *respexe34.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Você deverá digitar dois números diferentes de zero
Digite o primeiro número: 10
Digite o segundo número: 5

A soma 10 + 5 = 15
A subtração 10 - 5 = 5
O produto 10 x 5 = 50
A divisão 10 : 5 = 2
A média entre 10 e 5 = 7.50
  
```

Figura 3.9 – Resultado da execução do código do programa em C: *respexe34.cpp*

Fonte: Elaboração própria

3.5 Escreva um programa que solicite ao usuário um número inteiro e imprima a tabuada de multiplicação desse número até 6, sendo que o comando da operação de multiplicação deve ser usado dentro do printf.

Resposta: o código do programa é apresentado na Figura 3.10 e o resultado da execução desse código na Figura 3.11.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração das variáveis
6     int auxnum;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    // Início da lógica de programação
12    printf("**** Tabuada ****\n\n");
13    printf("Digite um número: ");
14    scanf("%d", &auxnum);
15    // Escrevendo Mensagens na Tela
16    printf("\n\n\t%d x 1 = %d\n",auxnum,auxnum*1);
17    printf("\t%d x 2 = %d\n",auxnum,auxnum*2);
18    printf("\t%d x 3 = %d\n",auxnum,auxnum*3);
19    printf("\t%d x 4 = %d\n",auxnum,auxnum*4);
20    printf("\t%d x 5 = %d\n",auxnum,auxnum*5);
21    printf("\t%d x 6 = %d\n",auxnum,auxnum*6);
22    getch(); //Função que aguarda o recebimento de um caractere via teclado
23 }
  
```

Figura 3.10 – Código do programa em C: *respexe35.cpp*

Fonte: Elaboração própria

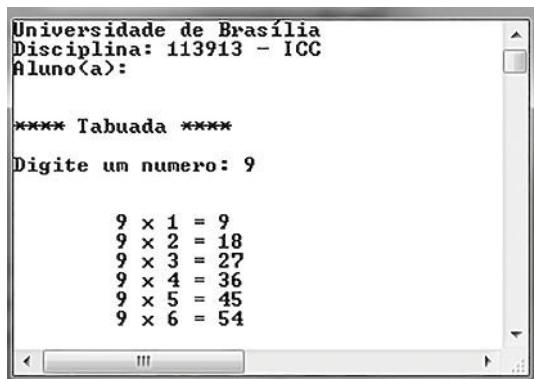


Figura 3.11 – Resultado da execução do código do programa em C: *respexe35.cpp*

Fonte: Elaboração própria

Capítulo 4

Estrutura de Controle de Fluxo Seleção Simples

Seleção Simples

4.1 Desenvolva um programa para comparar a idade de Pedro e a de Joana e informar quem é o mais velho.
 Dados de entrada: idade de Pedro e de Joana (tipo das variáveis: inteiro, e valor em anos). Observação: essas pessoas possuem idades diferentes.

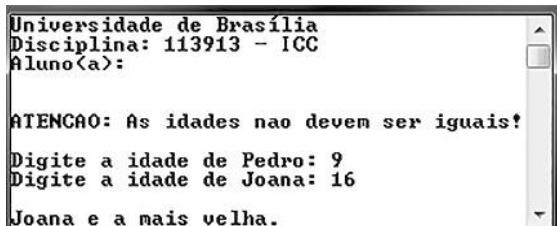
Resposta: o código do programa é apresentado na Figura 4.1 e o resultado da execução desse código na Figura 4.2 e 4.3.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de variáveis:
6     int iddP,iddJ;
7     //Cabeçalho:
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("ATENÇÃO: As idades não devem ser iguais!\n\n");
13    printf("Digite a idade de Pedro: ");
14    //Recebendo o valor da variável 'iddP':
15    scanf("%d",&iddP);
16    printf("Digite a idade de Joana: ");
17    //Recebendo o valor da variável 'iddJ':
18    scanf("%d",&iddJ);
19    //Verificação de quem é o mais velho:
20    if(iddP > iddJ)
21        printf("\nPedro é o mais velho.");
22    if(iddJ > iddP)
23        printf("\nJoana é a mais velha.");
24    getch(); //Função que aguarda o recebimento de um caractere via teclado.
25 }
```

Figura 4.1 – Código do programa em C: *respexe41.cpp*

Fonte: Elaboração própria

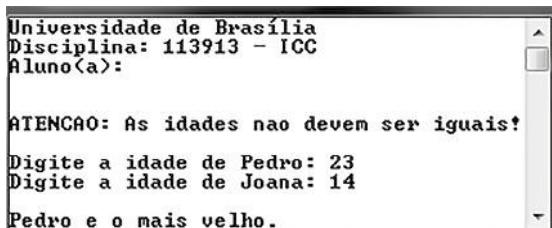


```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

ATENCAO: As idades nao devem ser iguais!
Digite a idade de Pedro: 9
Digite a idade de Joana: 16
Joana e a mais velha.
```

Figura 4.2 – Resultado da execução do código do programa em C: *respexe41.cpp*, que ilustra um exemplo em que a Joana é mais velha do que o Pedro

Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

ATENCAO: As idades nao devem ser iguais!
Digite a idade de Pedro: 23
Digite a idade de Joana: 14
Pedro e o mais velho.
```

Figura 4.3 – Resultado da execução do código do programa em C: *respexe41.cpp*, que ilustra um exemplo em que o Pedro é mais velho que a Joana

Fonte: Elaboração própria

4.2 Desenvolva um programa para calcular e para comparar a área de dois retângulos A e B. O programa deverá informar qual retângulo possui a maior área ou se possuem tamanhos iguais. Dados de entrada: tamanho da base e da altura (tipo das variáveis: inteiro, valor em centímetros).

Resposta: o código do programa é apresentado na Figura 4.4 e o resultado da execução desse código nas Figuras 4.5, 4.6 e 4.7.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de Variáveis
6     float area1, area2, base, altura;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Digite a base do primeiro retângulo em cm: ");
13    scanf("%f", &base);
14    printf("Digite a altura do primeiro retângulo em cm: ");
15    scanf("%f", &altura);
16    area1=altura*base;
17    printf("Digite a base do segundo retângulo em cm: ");
18    scanf("%f", &base);
19    printf("Digite a altura do segundo retângulo em cm: ");
20    scanf("%f", &altura);
21    area2=altura*base;
22    //Resultados
23    printf("\n\nA área do primeiro retângulo: %.2f cm\n", area1);
24    printf("A área do segundo retângulo: %.2f cm\n", area2);
25    //Comparação das Áreas
26    if(area1>area2)
27        printf("\nA área do primeiro retângulo é maior.\n");
28    if(area1<area2)
29        printf("\nA área do segundo retângulo é maior.\n");
30    if(area1==area2)
31        printf("\nAs áreas dos dois retângulos são iguais.\n");
32    getch(); //Função que aguarda o recebimento de um caractere via teclado.
33 }

```

Figura 4.4 – Código do programa em C: *respexe42.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite a base do primeiro retângulo em cm: 10
Digite a altura do primeiro retângulo em cm: 3
Digite a base do segundo retângulo em cm: 9
Digite a altura do segundo retângulo em cm: 12

A área do primeiro retângulo: 30.00 cm
A área do segundo retângulo: 108.00 cm

A área do segundo retângulo é maior.

```

Figura 4.5 – Resultado da execução do programa em C: *respexe42.cpp*, que ilustra um exemplo em que a área do segundo retângulo é maior que o primeiro

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite a base do primeiro retangulo em cm: 10
Digite a altura do primeiro retangulo em cm: 15
Digite a base do segundo retangulo em cm: 15
Digite a altura do segundo retangulo em cm: 20

A area do primeiro retangulo: 150.00 cm
A area do segundo retangulo: 300.00 cm

A area do segundo retangulo e maior.

```

Figura 4.6 – Resultado do programa em C: *respexe42.cpp*, que ilustra um outro exemplo em que a área do segundo retângulo é maior que a do primeiro

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite a base do primeiro retangulo em cm: 3
Digite a altura do primeiro retangulo em cm: 4
Digite a base do segundo retangulo em cm: 4
Digite a altura do segundo retangulo em cm: 3

A area do primeiro retangulo: 12.00 cm
A area do segundo retangulo: 12.00 cm

As areas do dois retangulos sao iguais.

```

Figura 4.7 – Resultado do programa em C: *respexe42.cpp*, que ilustra um exemplo em que os dois retângulos possuem áreas iguais

Fonte: Elaboração própria

- 4.3 Escreva um programa que leia a quantidade de alunas e de alunos. Depois esse programa deve informar se essa turma possui mais alunos ou mais alunas. Se essa turma possuir a quantidade de alunas maior que a de alunos, informe o total de alunos dessa turma. O programa deve verificar se a quantidade de alunos é igual a de alunas.

Resposta: o código do programa é apresentado na Figura 4.8 e o resultado da execução desse código nas Figuras 4.9 e 4.10.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main () //Programa principal
5     //Declaração de Variáveis
6     int qnt_alunos, qnt_alunas; // Variáveis do tipo inteiro para numero de alunos e alunas.
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n\n");
11    //Início da lógica de programação
12    printf("Digite a quantidade de alunos: ");
13    scanf("%d", &qnt_alunos); // Recebe via teclado a quantidade de alunos
14    printf("Digite a quantidade de alunas: ");
15    scanf("%d", &qnt_alunas); // Recebe via teclado a quantidade de alunas
16
17    if(qnt_alunos>qnt_alunas) // Teste que verifica se a nº alunos é maior que o de alunas
18        printf("\nExistem mais alunos do que alunas.\n");
19    if(qnt_alunas>qnt_alunos) // Teste que verifica se a nº alunas é maior que o de alunos
20        printf("\nExistem mais alunas do que alunos. O total de alunos é %d", qnt_alunas+qnt_alunos);
21    if(qnt_alunos==qnt_alunas) // // Teste que verifica se a nº alunos é igual ao de alunas
22        printf("\nO número de alunos e igual ao número de alunas: %d e %d", qnt_alunas, qnt_alunos);
23    getch()// Função para congelar a tela até que seja pressionado alguma tecla.
24

```

Figura 4.8 – Código do programa em C: *respexe43.cpp*

Fonte: Elaboração própria

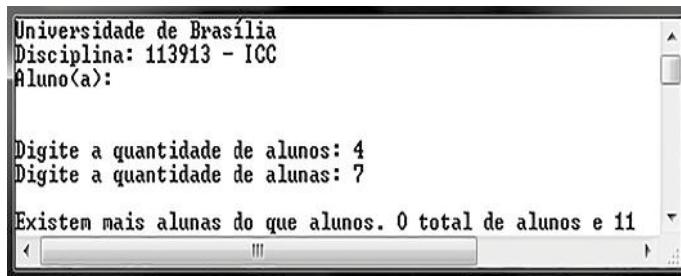


Figura 4.9 – Resultado da execução do código do programa em C: *respexe43.cpp*, que ilustra um exemplo em que a quantidade de alunas é maior que a quantidade de alunos

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite a quantidade de alunos: 15
Digite a quantidade de alunas: 10

Existem mais alunos do que alunas.

```

Figura 4.10 – Resultado do programa em C: *respexe43.cpp*, que ilustra um exemplo em que a quantidade de alunos é maior que a quantidade de alunas

Fonte: Elaboração própria

- 4.4 Faça um programa que receba um número inteiro e que verifique se esse número é par ou ímpar.

Resposta: o código do programa é apresentado na Figura 4.11 e o resultado da execução desse código nas Figuras 4.12 e 4.13.

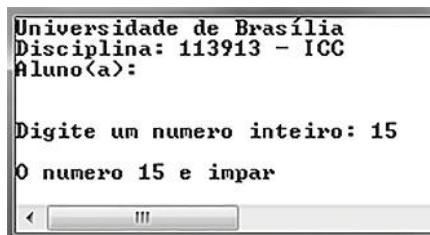
```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de Variáveis:
6     int num;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //escrevendo o comando na tela
12    printf("Digite um numero inteiro: ");
13    //recebendo do teclado o número inteiro
14    scanf("%d",&num);
15    //Verificando se o numero é par ou ímpar
16    if(num%2==0)
17        printf("\nO numero %d é par\n\n",num);
18    else
19        printf("\nO numero %d é ímpar\n\n",num);
20    getch(); //Função que aguarda o recebimento de um caractere via teclado.
21 }

```

Figura 4.11 – Código do programa em C: *respexe44.cpp*

Fonte: Elaboração própria

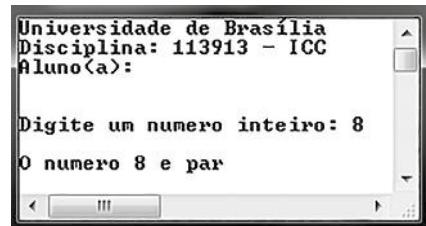


```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite um numero inteiro: 15
O numero 15 e impar
```

Figura 4.12 – Resultado da execução do código do programa em C: *respexe44.cpp*, que ilustra um exemplo em que o número digitado é ímpar

Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite um numero inteiro: 8
O numero 8 e par
```

Figura 4.13 – Resultado do programa em C: *respexe44.cpp*, que ilustra um exemplo em que o número digitado é par

Fonte: Elaboração própria

4.5 Escreva um programa em C que calcule o volume de um cilindro circular, dados o seu raio e sua altura. Observação: $V = \pi \times r^2 \times h$, em que $\pi=3,14$, r = raio e h = altura.

Resposta: o código do programa é apresentado na Figura 4.14 e o resultado da execução desse código na Figura 4.15.

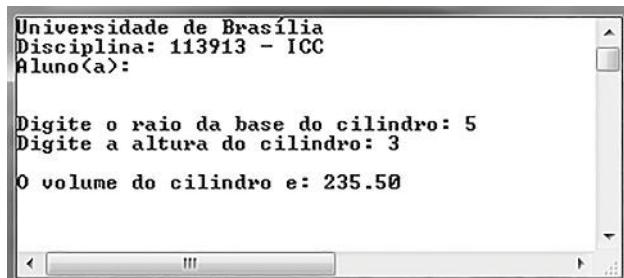
```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de Variáveis:
6     float raio, alt, vol;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11
12    //escrevendo o comando na tela
13    printf("Digite o raio da base do cilindro: ");
14    scanf("%f",&raio);//recebendo do teclado o raio
15    printf("Digite a altura do cilindro: ");
16    scanf("%f",&alt);//recebendo do teclado a altura
17    //Calculando o volume
18    vol=(3.14*((raio*raio)*alt));
19    /*Dica: abusem dos parênteses pois o compilador realiza o cálculo
20     em sequência não priorizando multiplicação e divisão.
21     Neste caso não é necessário, mas tomar como hábito é importante.*/
22    //imprimindo o volume na tela
23    printf("\nO volume do cilindro é: %.2f",vol);
24    getch(); //Função que aguarda o recebimento de um caractere via teclado.
25}

```

Figura 4.14 – Código do programa em C: *respexe45.cpp*

Fonte: Elaboração própria

Figura 4.15 – Resultado da execução do código do programa em C: *respexe45.cpp*

Fonte: Elaboração própria

- 4.6 Desenvolva um programa para comparar a idade de Pedro, de Joana e de Ismael e informar quem é o mais velho. Dados de entrada: idade de Pedro, de Joana e de Ismael (tipo das variáveis: inteiro, e valor em anos). Considere que essas pessoas possuem idades diferentes.

Resposta: o código do programa é apresentado na Figura 4.16 e o resultado da execução desse código nas Figuras 4.17 e 4.18.

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  int main (){//Programa principal
5      //Declaração de Variáveis:
6      int iddPedro, iddIsmael, iddJoana;
7      //Cabeçalho
8      printf("Universidade de Brasília\n", 161);
9      printf("Disciplina: 113913 - ICC \n");
10     printf("Aluno(a):\n\n");
11     //Início da lógica de programação
12     printf("Digite a idade de Pedro: ");
13     scanf("%d",&iddPedro);
14     printf("Digite a idade de Ismael: ");
15     scanf("%d",&iddIsmael);
16     printf("Digite a idade de Joana: ");
17     scanf("%d",&iddJoana);
18     //Estrutura condicional para avaliação de quem é mais velho dentre os três
19     if((iddPedro>iddIsmael)&&(iddPedro>iddJoana))
20         printf("\n\nPedro é o mais velho dentre os tres\n\n");
21     else {
22         if(iddIsmael>iddJoana)
23             printf("\n\nIsmael é o mais velho dentre os tres\n\n");
24         //este "else" pertence ao segundo "if"
25         else
26             printf("\n\nJoana é a mais velha dentre os tres\n\n");
27     }
28     getch(); //Função que aguarda o recebimento de um caractere via teclado.
29 }
```

Figura 4.16 – Código do programa em C: *respexe46.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Digite a idade de Pedro: 19
Digite a idade de Ismael: 20
Digite a idade de Joana: 15

Ismael é o mais velho dentre os tres

```

Figura 4.17 – Resultado da execução do código do programa em C: *respexe46.cpp*, que ilustra um exemplo em que Ismael é o mais velho

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Digite a idade de Pedro: 13
Digite a idade de Ismael: 5
Digite a idade de Joana: 7

Pedro é o mais velho dentre os tres
```

Figura 4.18 – Resultado da execução do código do programa em C: *respexe46.cpp*, que ilustra um exemplo em que Pedro é o mais velho

Fonte: Elaboração própria

- 4.7 Faça um programa que receba um número inteiro e que verifique se esse número é par ou ímpar. O programa deve informar:
- se o número é par, caso afirmativo informar também se ele é maior que 15;
 - se o número é ímpar, caso afirmativo se ele é menor que 50.

Resposta: o código do programa é apresentado na Figura 4.19 e o resultado da execução desse código nas Figuras 4.20 e 4.21.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h>/Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de variáveis:
6     int auxnum;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Digite um numero: ");
13    scanf("%d",&auxnum);//Recebendo o valor da variável 'auxnum';
14    //Condição = Se o 'auxnum' dividido por 2 retornar o valor '0', ele será par:
15    if(auxnum%2 == 0)
16    {
17        printf("\n\nO numero digitado é par.\n");
18        //Condição extra para verificar se o numero é maior que 15:
19        if(auxnum > 15)
20            printf("O numero digitado tambem é maior que 15.\n");
21    }
22    //Condição = Se o 'auxnum' dividido por 2 retornar o valor '1', ele será ímpar:
23    if(auxnum%2 == 1)
24    {
25        printf("\n\nO numero digitado é ímpar.\n");
26        //Condição extra para verificar se o numero é menor que 50:
27        if(auxnum < 50)
28            printf("O numero digitado tambem é menor que 50.\n");
29    }
30    getch(); //Função que aguarda o recebimento de um caractere via teclado.
31 }
```

Figura 4.19 – Código do programa em C: *respexe47.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite um numero: 20

O numero digitado é par.
O numero digitado tambem é maior que 15.

```

Figura 4.20 – Resultado da execução do código do programa em C: *respexe47.cpp*, que ilustra um exemplo em que o numero digitado é par e maior que 15

Fonte: Elaboração própria

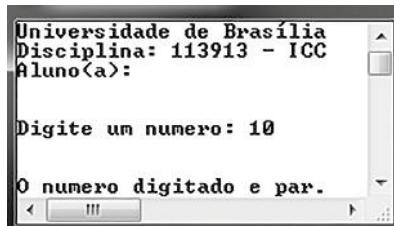


Figura 4.21 – Resultado da execução do código do programa em C: *respexe47.cpp*, que ilustra um exemplo em que número digitado é par e menor que 15

Fonte: Elaboração própria

4.8 Faça um programa que receba um número inteiro e que verifique se esse número é par ou se é ímpar. O programa deve informar ao usuário o número apenas se for par e maior que 15 e se for ímpar apenas se for menor que 50.

Resposta: o código do programa é apresentado na Figura 4.22 e o resultado da execução desse código nas Figuras 4.23 e 4.24.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h>/Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de variáveis:
6     int auxnum,resp_bool=0;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Digite um numero: ");
13    scanf("%d",&auxnum);
14    //Condição = Se o 'num' dividido por 2 retornar o valor '0', ele será par:
15    if(auxnum%2 == 0 && auxnum > 15){
16        printf("\n\nO numero digitado é par e maior que 15.\n");
17        resp_bool=1; //Variável auxiliar assume valor lógico positivo = 1
18    }
19    //Condição = Se o 'num' dividido por 2 retornar o valor '1', ele será ímpar:
20    if(auxnum%2 == 1 && auxnum < 50){
21        printf("\n\nO numero digitado é ímpar.\n");
22        resp_bool=1; //Variável auxiliar assume valor lógico positivo = 1
23    }
24    //Se a variável 'resp_bool' for '0', quer dizer que nenhuma das condições acima foi satisfeita:
25    if(resp_bool == 0)
26        printf("\n\nO numero digitado não entra em nenhuma condição do programa.\n");
27    getch(); //Função que aguarda o recebimento de um caractere via teclado.
28 }
```

Figura 4.22 – Código do programa em C: *respexe48.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite um numero: 45

O numero digitado e impar.

```

Figura 4.23 – Resultado da execução do código do programa em C: *respexe48.cpp*, que ilustra um exemplo em que o número digitado é ímpar e menor que 50

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite um numero: 18

O numero digitado e par e maior que 15.

```

Figura 4.24 – Resultado da execução do código do programa em C: *respexe48.cpp*, que ilustra um outro exemplo em que o número digitado é ímpar e menor que 50

Fonte: Elaboração própria

- 4.9 Escreva um programa em C que calcule o volume de dois cilindros circulares e que verifique qual deles é maior. O raio e altura dos cilindros serão informados pelo usuário. Observação: $V = \pi r^2 h$, em que $\pi = 3,14$, $r =$ raio e $h =$ altura.

Resposta: o código do programa é apresentado na Figura 4.25 e o resultado da execução desse código nas Figuras 4.26, 4.27 e 4.28.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de variáveis:
6     float vol1, vol2, alt, raio,pi=3.14;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Digite o raio do primeiro cilindro: ");
13    scanf("%f",&raio);
14    printf("Digite a altura do primeiro cilindro: ");
15    scanf("%f",&alt);
16    //Calculo do volume 1
17    vol1=pi*(raio*raio)*alt;
18    printf("Digite o raio do segundo cilindro: ");
19    scanf("%f",&raio);
20    printf("Digite a altura do segundo cilindro: ");
21    scanf("%f",&alt);
22    //Calculo do volume 2
23    vol2=pi*(raio*raio)*alt;
24    //Resultados:
25    printf("\n\nO volume do cilindro 1 é: %.2f\n",vol1);
26    printf("O volume do cilindro 2 é: %.2f\n\n",vol2);
27    //Verificação do maior volume:
28    if(vol1 > vol2)
29        printf("O volume do cilindro 1 é maior que o volume do cilindro 2.\n");
30    if(vol2 > vol1)
31        printf("O volume do cilindro 2 é maior que o volume do cilindro 1.\n");
32    if(vol2 == vol1)
33        printf("Os volumes dos cilindros são iguais.\n");
34    getch(); //Função que aguarda o recebimento de um caractere via teclado.
35 }

```

Figura 4.25 – Código do programa em C: *respexe49.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o raio do primeiro cilindro: 9
Digite a altura do primeiro cilindro: 2
Digite o raio do segundo cilindro: 10
Digite a altura do segundo cilindro: 4

O volume do cilindro 1 é: 508.68
O volume do cilindro 2 é: 1256.00

O volume do cilindro 2 é maior que o volume do cilindro 1.

```

Figura 4.26 – Resultado da execução do código do programa em C: *respexe49.cpp*, que ilustra um exemplo em que o volume do cilindro 2 é maior que do cilindro 1

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite o raio do primeiro cilindro: 12
Digite a altura do primeiro cilindro: 4
Digite o raio do segundo cilindro: 5
Digite a altura do segundo cilindro: 3

O volume do cilindro 1 e: 1808.64
O volume do cilindro 2 e: 235.50

O volume do cilindro 1 e maior que o volume do cilindro 2.
  
```

Figura 4.27 – Resultado da execução do código do programa em C: *respexe49.cpp*, que ilustra um exemplo em que o volume do cilindro 1 é maior que do cilindro 2

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite o raio do primeiro cilindro: 3
Digite a altura do primeiro cilindro: 4
Digite o raio do segundo cilindro: 3
Digite a altura do segundo cilindro: 4

O volume do cilindro 1 e: 113.04
O volume do cilindro 2 e: 113.04

Os volumes dos cilindros sao iguais.
  
```

Figura 4.28 – Resultado da execução do código do programa em C: *respexe49.cpp*, que ilustra um exemplo em que o volume dos cilindros 1 e 2 são iguais

Fonte: Elaboração própria

- 4.10 Escreva um programa que informe o valor de uma corrida de taxi. Para calcular o valor da corrida é necessário saber a distância percorrida em quilômetros e qual o tipo da bandeira da corrida, 1 ou 2. Caso a bandeira seja 1, o preço do quilômetro percorrido é de R\$ 1,80, se a bandeira for 2 o valor é de R\$ 2,30. Escreva um programa em linguagem C que solicite a

distância percorrida em quilômetros e qual o tipo da bandeira da corrida e informe o valor da corrida.

Resposta: o código do programa é apresentado na Figura 4.29 e o resultado da execução desse código nas Figuras 4.30 e 4.31.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de variáveis:
6     int band;
7     float kms,valor,valor_corrida;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n\n");
12    //Início da lógica de programação
13    printf("Digite o numero da bandeira (1 ou 2): ");
14    scanf("%d",&band);
15    printf("Digite a kilometragem rodada: ");
16    scanf("%f",&kms);
17    //Definição dos preços/km dependendo da bandeira:
18    if(band == 1)
19        valor = 1.8;
20    if(band == 2)
21        valor = 2.3;
22    //Calculo do valor da corrida:
23    valor_corrida = valor*kms;
24    //Resultados:
25    printf("O valor da corrida e: R$ %.2f\n", valor_corrida);
26    getch(); //Função que aguarda o recebimento de um caractere via teclado.
27 }
```

Figura 4.29 – Código do programa em C: *respexe410.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o numero da bandeira (1 ou 2): 2
Digite a kilometragem rodada: 34
O valor da corrida e: R$ 78.20

```

Figura 4.30 – Resultado da execução do código do programa em C: *respexe410.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira 2, e a quantidade de quilômetros rodados foi 34

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite o numero da bandeira (1 ou 2): 1
Digite a kilometragem rodada: 23
O valor da corrida e: R$ 41.40
```

Figura 4.31 – Resultado do programa em C: *respexe410.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira 1, e a quantidade de quilômetros rodados foi 23

Fonte: Elaboração própria

- 4.11 Escreva um programa para informar o valor de uma corrida de taxi que considere o valor mínimo da corrida. Ou seja, ao entrar no taxi, o cliente deve pagar um valor mínimo de R\$ 3,50, mesmo que o tamanho da corrida seja inferior a esse valor. O programa deve considerar as informações do exercício anterior.

Resposta: o código do programa é apresentado na Figura 4.32 e o resultado da execução desse código nas Figuras 4.33 e 4.34.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de variáveis:
6     int band;
7     float kms,valor,valor_corrida,min=3.5;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12    //Início da lógica de programação
13    printf("ATENCAO: Taxa minima do taxi: %.2f\n",min);
14    printf("Digite o numero da bandeira: ");
15    scanf("%d",&band);
16    printf("Digite a kilometragem rodada: ");
17    scanf("%f",&kms);
18    //Definição dos preços/km dependendo da bandeira:
19    if(band == 1)
20        valor = 1.8;
21    if(band == 2)
22        valor = 2.3;
23    //Calculo do valor da corrida com a soma do minimo da corrida
24    valor_corrida = valor*kms;
25    if(valor_corrida<min);
26        valor_corrida=min;
27    //Resultados:
28    printf("O valor da corrida e: R$ %.2f\n", valor_corrida);
29    getch(); //Função que aguarda o recebimento de um caractere via teclado.
30 }

```

Figura 4.32 – Código do programa em C: *respexe411.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

ATENCAO: Taxa minima do taxi: 3.50
Digite o numero da bandeira: 1
Digite a kilometragem rodada: 0.5
O valor da corrida e: R$ 3.50

```

Figura 4.33 – Resultado da execução do programa em C: *respexe411.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira 1, e a quantidade de quilômetros rodados foi 0,5

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
ATENCAO: Taxa minima do taxi: 3.50
Digite o numero da bandeira: 1
Digite a kilometragem rodada: 14
O valor da corrida e: R$ 25.20
```

Figura 4.34 – Resultado do programa em C: *respexe411.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira 1, e a quantidade de quilômetros rodados foi 14

Fonte: Elaboração própria

- 4.12 Escreva um programa que informe o valor de uma corrida de taxi em Brasília. A diferença em relação ao exercício anterior é que alguns taxis oferecem um desconto de 30%. Altere o programa do exercício anterior para que solicite se a corrida possui ou não desconto de 30% (1 – sim, 2 – não).

Resposta: o código do programa é apresentado na Figura 4.35 e o resultado da execução desse código na Figura 4.36.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de variáveis:
6     int band,desc;
7     float kms,valor,valor_corrida,min=3.5;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12    //Início da lógica de programação
13    printf("Digite o numero da bandeira (1 ou 2): ");
14    scanf("%d",&band);
15    printf("Digite a kilometragem rodada: ");
16    scanf("%f",&kms);
17    //Definição dos preços/km dependendo da bandeira:
18    if(band == 1)
19        valor = 1.8;
20    if(band == 2)
21        valor = 2.3;
22    //Calculo do valor da corrida com a soma do minimo da corrida:
23    valor_corrida = valor*kms;
24    //Verificação do desconto:
25    printf("O taxi oferece desconto de 30% ? (1 - SIM ou 2 - NAO)\n");
26    scanf("%d",&desc);
27    if(desc == 1)
28        valor_corrida = valor_corrida*0.7;
29    if(valor_corrida<min)
30        valor_corrida=min;
31    //Resultados:
32    printf("O valor da corrida é: R$ %.2f\n", valor_corrida);
33    getch(); //Função que aguarda o recebimento de um caractere via teclado.
34 }

```

Figura 4.35 – Código do programa em C: *respexe412.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o numero da bandeira (1 ou 2): 2
Digite a kilometragem rodada: 10
O taxi oferece desconto de 30% ? (1 - SIM ou 2 - NAO)
1
O valor da corrida é: R$ 16.10

```

Figura 4.36 – Resultado da execução do código do programa em C: *respexe412.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira 2, a quantidade de quilômetros rodados foi 10, e o desconto foi de 30%

Fonte: Elaboração própria

Capítulo 5

**Estrutura de Controle de Fluxo
Condisional Composta**

Condisional Composta

- 5.1 Escreva um programa que leia o número de alunos e o de alunas de uma sala. Como saída, o programa deve apresentar primeiro quem estiver em maior quantidade. Por exemplo, se na sala tiver mais alunos, apresente primeiro o número de alunos, caso contrário apresente o número de alunas e depois o de alunos. Considere o caso em que o número de alunos é igual ao número de alunas.

Resposta: o código do programa é apresentado na Figura 5.1 e o resultado da execução desse código nas Figuras 5.2, 5.3 e 5.4.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de Variáveis:
6     int alunos, alunas;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    // Início do programa
12    printf("Digite o numero de alunos ");
13    scanf("%d", &alunos);
14    printf("Digite o numero de alunas ");
15    scanf("%d", &alunas);
16
17    // Apresentação dos resultados como solicitado utilizando estrutura
18    // condicional composta:
19    if(alunos>alunas)
20    {
21        printf("\n\nO numero de alunos = %d\n", alunos);
22        printf("O numero de alunas = %d\n", alunas);
23    }
24    else if (alunas>alunos)
25    {
26        printf("\n\nO numero de alunas = %d\n", alunas);
27        printf("O numero de alunos = %d\n", alunos);
28    }
29    else
30        printf("\n\nO numero de alunas e alunos são iguais");
31    getch(); //Função que aguarda o recebimento de um caractere via teclado.
32 }
```

Figura 5.1 – Código do programa em C: *respexe51.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o numero de alunos 14
Digite o numero de alunas 16

O numero de alunas = 16
O numero de alunos = 14
```

Figura 5.2 – Resultado da execução do código programa em C: *respexe51.cpp*, que ilustra o caso em que a quantidade de alunas é maior que a quantidade de alunos

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o numero de alunos 23
Digite o numero de alunas 18

O numero de alunos = 23
O numero de alunas = 18
```

Figura 5.3 – Resultado da execução do código programa em C: *respexe51.cpp*, que ilustra um exemplo em que a quantidade de alunos é maior que a quantidade de alunas

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o numero de alunos 15
Digite o numero de alunas 15

O numero de alunas e alunos sao iguais
```

Figura 5.4 – Resultado da execução do código programa em C: *respexe51.cpp*, que ilustra um exemplo em que a quantidade de alunas e de alunos são iguais

Fonte: Elaboração própria

- 5.2 Desenvolva um programa para comparar a idade de Pedro e de Joana e informar quem é o mais velho. Dados de entrada: idade de Pedro e de Joana (tipo das variáveis: inteiro, e valor em anos). Observação: essas pessoas possuem idades diferentes.

Resposta: o código do programa é apresentado na Figura 5.5 e o resultado da execução desse código nas Figuras 5.6 e 5.7.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de Variáveis:
6     int iddPedro, iddJoana;
7     //Cabecalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n\n");
11    // Início do programa
12    printf("Digite a idade de Pedro ");
13    scanf("%d", &iddPedro);
14    printf("Digite a idade de Joana ");
15    scanf("%d", &iddJoana);
16
17    // Apresentação dos resultados como solicitado utilizando estrutura
18    // condicional composta:
19    if(iddPedro>iddJoana)
20        printf("\nPedro é o mais velho");
21    else if (iddJoana>iddPedro)
22        printf("\nJoana é a mais velha");
23    else
24        printf("\nJoana e Pedro tem a mesma idade");
25    getch(); //Função que aguarda o recebimento de um caractere via teclado.
26 }
```

Figura 5.5 – Código do programa em C: *respexe52.cpp*

Fonte: Elaboração própria



Figura 5.6 – Resultado da execução do código programa em C: *respexe52.cpp*, que ilustra um exemplo em que Joana é mais velha

Fonte: Elaboração própria



Figura 5.7 – Resultado da execução do código programa em C: *respexe52.cpp*, que ilustra um exemplo em que Pedro é mais velho

Fonte: Elaboração própria

5.3 Desenvolva um programa para calcular e comparar a área de dois retângulos A e B. O programa deverá dizer qual retângulo possui a maior área ou se ambos possuem tamanhos iguais. Dados de entrada: tamanho da base e da altura (tipo das variáveis: inteiro, valor em centímetros).

Resposta: o código do programa é apresentado na Figura 5.8 e do resultado da execução desse código nas Figuras 5.9 e 5.10.

```

1 #include<stdio.h> //declarando biblioteca de entrada/saída
2 #include<conio.h> //declarando biblioteca para uso de funções auxiliares
3 #include<stdlib.h>/declarando biblioteca para uso de funções auxiliares
4 int main(){ //Programa principal
5     //Declaração de Variáveis:
6     int altura, base, area1, area2;
7     //Cabecalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    // Início da lógica de programação
12    printf("Digite a altura do retângulo 1: ");
13    scanf("%d", &altura);
14    printf("Digite a Base do retângulo 1: ");
15    scanf("%d", &base);
16    area1=altura*base; // Calculando área do retângulo 1
17    printf("\nDigite a altura do retângulo 2: ");
18    scanf("%d", &altura);
19    printf("Digite a Base do retângulo 2: ");
20    scanf("%d", &base);
21    area2=altura*base; // Calculando área do retângulo 2
22
23    // Apresentando resultados na tela utilizando estrutura condicional composta
24    // como solicitado pela questão
25    if(area1>area2)
26    {
27        printf("\nA área do retângulo 1 é maior do que a área do retângulo 2\n");
28        printf("Área do retângulo 1 = %d\n", area1);
29        printf("Área do retângulo 2 = %d\n", area2);
30    }
31    else if(area2>area1)
32    {
33        printf("\nA área do retângulo 2 é maior do que a área do retângulo 1\n");
34        printf("Área do retângulo 2 = %d\n", area2);
35        printf("Área do retângulo 1 = %d\n", area1);
36    }
37    else if(area1==area2)
38    {
39        printf("\nAs áreas dos retângulos são iguais e correspondem a: %d\n", area1);
40        getch(); //Função que aguarda o recebimento de um caractere via teclado.
}

```

Figura 5.8 – Código do programa em C: *respexe53.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite a altura do retângulo 1: 3
Digite a Base do retângulo 1: 4

Digite a altura do retângulo 2: 8
Digite a Base do retângulo 2: 5

A área do retângulo 2 é maior do que a área do retângulo 1
Área do retângulo 2 = 40
Área do retângulo 1 = 12

```

Figura 5.9 – Resultado da execução do código programa em C: *respexe53.cpp*, que ilustra um exemplo em que a área do retângulo 2 é maior que do retângulo 1

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Digite a altura do retangulo 1: 5
Digite a Base do retangulo 1: 2

Digite a altura do retangulo 2: 1
Digite a Base do retangulo 2: 6

A area do retangulo 1 é maior do que a area do retangulo 2
Area do retangulo 1 = 10
Area do retangulo 2 = 6
```

Figura 5.10 – Resultado da execução do código programa em C: *respexe53.cpp*, que ilustra um exemplo em que a área do retângulo 1 é maior que a do retângulo 2

Fonte: Elaboração própria

5.4 Faça um programa que receba dois números inteiros e que informe o maior deles. O programa deve verificar também se o maior número é par ou ímpar.

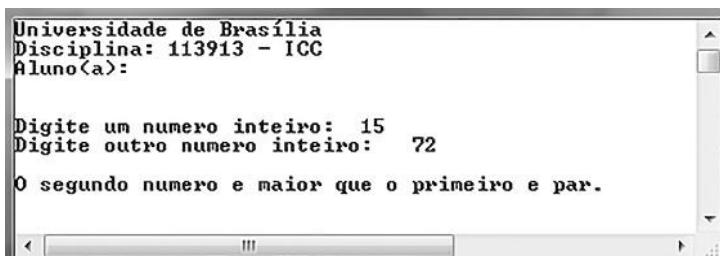
Resposta: o código do programa é apresentado na Figura 5.11 e o resultado da execução desse código nas Figuras 5.12 e 513.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h>/Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de Variáveis:
6     int num1, num2, resto;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Digite um numero inteiro: ");
13    scanf("%d", &num1);
14    printf("Digite outro numero inteiro: ");
15    scanf("%d", &num2);
16    //Condições para verificação do maior dos dois números e sua paridade
17    if (num1>num2)
18    {
19        resto=num1%2;
20        if (resto==0)
21            printf("\nO primeiro numero é maior que o segundo e par.");
22        else
23            printf("\nO primeiro numero é maior que o segundo e impar.");
24    }
25    else if (num2>num1)
26    {
27        resto=num2%2;
28        if (resto==0)
29            printf("\nO segundo numero é maior que o primeiro e par.");
30        else
31            printf("\nO segundo numero é maior que o primeiro e impar.");
32    }
33 } //Função que aguarda o recebimento de um caractere via teclado.
34 }
```

Figura 5.11 – Código do programa em C: *respexe54.cpp*

Fonte: Elaboração própria

Figura 5.12 – Resultado da execução do código do programa em C: *respexe54.cpp*, que ilustra um exemplo em que o segundo número é maior que o primeiro e é par

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  

Digite um numero inteiro: 17
Digite outro numero inteiro: 4
O primeiro numero e maior que o segundo e impar.

```

Figura 5.13 – Resultado da execução do código do programa em C: *respexe54.cpp*, que ilustra um exemplo em que o primeiro número é maior que o segundo e é par

Fonte: Elaboração própria

5.5 Uma frutaria vende frutas com a seguinte tabela de preços:

Até 5 Kg	Acima de 5 Kg
Morango: R\$ 7,50 p/Kg	R\$ 5,30 p/Kg
Maçã: R\$ 3,50 p/Kg	R\$ 2,80 p/Kg

Se o cliente comprar menos de 5 kg de frutas e o valor total da compra ultrapassar R\$ 19,00, receberá um desconto de 8% sobre o total. Escreva um programa em C para ler a quantidade (em Kg) de morangos e a de maçãs (em Kg) e que informe o valor a ser pago pelo cliente.

Resposta: o código do programa é apresentado na Figura 5.14 e o resultado da execução desse código nas Figuras 5.15 e 5.16

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  int main (){//Programa principal
5      //Declaração de Variáveis:
6      float total, maca, morango, ttlmaca, ttlmorango, pesototal;
7      //Cabeçalho
8      printf("Universidade de Brasília\n", 161);
9      printf("Disciplina: 113913 - ICC \n");
10     printf("Aluno(a):\n\n\n");
11     //Início da lógica de programação
12     printf("Digite o peso total de macas em kg: ");
13     scanf("%f", &maca);
14     printf("Digite o peso total de morangos em kg: ");
15     scanf("%f", &morango);
16     pesototal=morango+maca;
17     // Condições para o preço baseado nos kg comprados
18     if (pesototal<5)
19     {
20         ttlmorango=7.5*morango;
21         ttlmaca=3.5*maca;
22     }
23     else
24     {
25         ttlmorango=5.3*morango;
26         ttlmaca=2.8*maca;
27     }
28     total=ttlmorango+ttlmaca; // Calculo do preço total
29     if (total>19)
30     {
31         total=0.92*total;
32         printf("\nO valor total da compra é: R$%.2f", total);
33     }
34     else
35         printf("\nO valor total da compra é: R$%.2f", total);
36     getch(); //Função que aguarda o recebimento de um caractere via teclado
37 }
```

Figura 5.14 – Código do programa em C: respexe55.cpp

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o peso total de macas em kg: 3.4
Digite o peso total de morangos em kg: 4.1
O valor total da compra e: R$28.75
  
```

Figura 5.15 – Resultado da execução do código do programa em C: *respexe55.cpp*, que ilustra um exemplo em que o peso das maçãs é de 3,4 quilos e o peso dos morangos é de 4,1 quilos

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o peso total de macas em kg: 5.4
Digite o peso total de morangos em kg: 1.3
O valor total da compra e: R$20.25
  
```

Figura 5.16 – Resultado da execução do código do programa em C: *respexe55.cpp*, que ilustra um exemplo em que o peso das maçãs é de 5,4 quilos e o peso dos morangos é de 1,3 quilos

Fonte: Elaboração própria

5.6 Desenvolva um programa para comparar a idade de Pedro, de Joana e de Ismael e informar quem é o mais velho. Dados de entrada: idade de Pedro, de Joana e de Ismael (tipo das variáveis: inteiro, e valor em anos). Considere que essas pessoas possuem idades diferentes.

Resposta: o código do programa é apresentado na Figura 5.17 e o resultado da execução desse código nas Figuras 5.18 e 5.19.

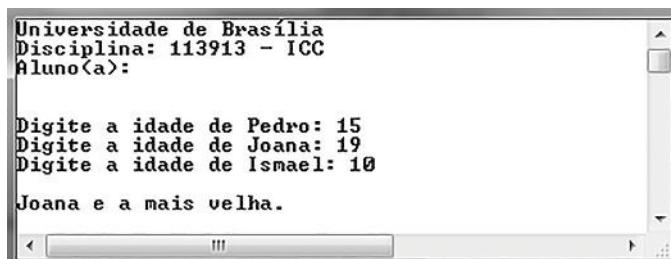
```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de Variáveis:
6     int iddPedro, iddJoana, iddIsmael;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("\nAluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Digite a idade de Pedro: ");
13    scanf("%d", &iddPedro);
14    printf("Digite a idade de Joana: ");
15    scanf("%d", &iddJoana);
16    printf("Digite a idade de Ismael: ");
17    scanf("%d", &iddIsmael);
18    //Condições
19    if (iddPedro>iddJoana && iddPedro>iddIsmael)
20        printf("\nPedro é o mais velho.");
21    else if (iddJoana>iddIsmael)
22        printf("\nJoana é a mais velha.");
23    else
24        printf("\nIsmael é o mais velho.");
25    getch(); //Função que aguarda o recebimento de um caractere via teclado.
26}

```

Figura 5.17 – Código do programa em C: *respexe56.cpp*

Fonte: Elaboração própria

Figura 5.18 – Resultado da execução do código do programa em C: *respexe56.cpp*, que ilustra um exemplo em que Joana é a mais velha

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):
Digite a idade de Pedro: 15
Digite a idade de Joana: 12
Digite a idade de Ismael: 13
Pedro é o mais velho.
```

Figura 5.19 – Resultado da execução do código do programa em C: *respexe56.cpp*, que ilustra um exemplo em que Pedro é o mais velho

Fonte: Elaboração própria

- 5.7 Faça um programa que receba um número inteiro e que verifique se esse número é par ou ímpar. O programa deve informar se o número é par, caso afirmativo informar também se é ou não maior que 15 ou se o número é ímpar, caso afirmativo informar se é ou não menor que 50.

Resposta: o código do programa é apresentado na Figura 5.20 e o resultado da execução desse código nas Figuras 5.21 e 5.22.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de Variáveis:
6     int num;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Digite um numero inteiro: ");
13    scanf("%d", &num);
14    //Condições e resultados
15    if ((num%2)==0)
16    {
17        if (num>15)
18            printf("\nO numero que voce digitou é par e maior que 15.");
19        else
20            printf("\nO numero que voce digitou é par e menor que 15.");
21    }
22    else
23    {
24        if (num<50)
25            printf("\nO numero que voce digitou é impar e menor que 50.");
26        else
27            printf("\n O numero que voce digitou é impar e maior que 50.");
28    }
29 }
30 getch(); //Função que aguarda o recebimento de um caractere via teclado.

```

Figura 5.20 – Código do programa em C: *respexe57.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite um numero inteiro: 15
O numero que voce digitou é impar e menor que 50.

```

Figura 5.21 – Resultado da execução do código do programa em C: *respexe57.cpp*, que ilustra um exemplo em que o número digitado é ímpar e menor que 50

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):
Digite um numero inteiro: 34
O numero que voce digitou e par e maior que 15.
```

Figura 5.22 – Resultado da execução do código do programa em C: *respexe57.cpp*, que ilustra um exemplo em que o número digitado é par e maior que 15

Fonte: Elaboração própria

- 5.8 Escreva um programa para informar o valor de uma corrida de taxi. Para calcular o valor da corrida é necessário saber a distância percorrida em quilômetros e qual o tipo da bandeira da corrida, 1 ou 2. Caso a bandeira seja 1, o preço do quilômetro rodado é de R\$ 1,80, se a bandeira for 2 o valor do quilômetro rodado é de R\$ 2,30. Escreva um programa em linguagem C que solicite a distância percorrida em quilômetros e qual o tipo da bandeira da corrida e informe o valor da corrida.

Resposta: o código do programa é apresentado na Figura 5.23 e o resultado da execução desse código nas Figuras 5.24 e 5.25.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h>/Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de variáveis:
6     int band;
7     float kms,valor_corrida;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n\n");
12    //Início da lógica de programação
13    printf("Digite o numero da bandeira (1 ou 2): ");
14    scanf("%d",&band);
15    printf("Digite a kilometragem rodada: ");
16    scanf("%f",&kms);
17    //Definição dos preços/km dependendo da bandeira:
18    if(band == 1)
19        valor_corrida = 1.8*kms; //Calculo do valor da corrida
20    else if(band == 2)
21        valor_corrida = 2.3*kms; //Calculo do valor da corrida
22    //Resultados:
23    printf("O valor da corrida e: R$ %.2f\n", valor_corrida);
24    getch(); //Função que aguarda o recebimento de um caractere via teclado
25 }

```

Figura 5.23 – Código do programa em C: *respexe58.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o numero da bandeira (1 ou 2): 2
Digite a kilometragem rodada: 19
O valor da corrida e: R$ 43.70

```

Figura 5.24 – Resultado da execução do código do programa em C: *respexe58.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira 2, e a quantidade de quilômetros rodados foi 19

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o numero da bandeira (1 ou 2): 1
Digite a kilometragem rodada: 10
O valor da corrida e: R$ 18.00

```

Figura 5.25 – Resultado da execução do código do programa em C: *respexe58.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira da corrida é 1, e a quantidade de quilômetros rodados foi 10

Fonte: Elaboração própria

- 5.9 Escreva um programa para informar o valor de uma corrida de taxi que considere o valor mínimo da corrida. Ou seja, ao entrar no taxi, o cliente deve pagar um valor mínimo de R\$ 3,50, independente do tamanho da corrida. O programa deve considerar as informações do exercício anterior.

Resposta: o código do programa é apresentado na Figura 5.26 e do resultado da execução desse código nas Figuras 5.27 e 5.28.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de variáveis:
6     int band;
7     float kms,valor_corrida,min=3.5;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n\n");
12    //Início da lógica de programação
13    printf("ATENÇÃO: Taxa mínima do taxi: %.2f\n",min);
14    printf("Digite o numero da bandeira: ");
15    scanf("%d",&band);
16    printf("Digite a kilometragem rodada: ");
17    scanf("%f",&kms);
18    //Definição dos preços/km dependendo da bandeira:
19    if(band == 1)
20        valor_corrida = 1.8*kms; //Calculo do valor da corrida
21    else if(band == 2)
22        valor_corrida = 2.3*kms; //Calculo do valor da corrida
23    if(valor_corrida<min)
24        valor_corrida=min;
25    //Resultados:
26    printf("O valor da corrida e: R$ %.2f\n", valor_corrida);
27    getch(); //Função que aguarda o recebimento de um caractere via teclado
28 }
```

Figura 5.26 – Código do programa em C: *respexe59.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

ATENCAO: Taxa minima do taxi: 3.50
Digite o numero da bandeira: 1
Digite a kilometragem rodada: 2
O valor da corrida e: R$ 3.60

```

Figura 5.27 – Resultado da execução do código do programa em C: *respexe59.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira 1, e a quantidade de quilômetros rodados foi 2

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

ATENCAO: Taxa minima do taxi: 3.50
Digite o numero da bandeira: 2
Digite a kilometragem rodada: 19
O valor da corrida e: R$ 43.70

```

Figura 5.28 – Resultado da execução do código do programa em C: *respexe59.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira 2, e a quantidade de quilômetros rodados foi 19

Fonte: Elaboração própria

- 5.10 Escreva um programa para informar o valor de uma corrida de taxi em Brasília. A diferença com o exercício anterior é que alguns taxis oferecem um desconto de 30 %. Altere o programa do exercício anterior para que ele solicite se a corrida possui ou não desconto de 30% (1 – sim, 2 – não).

Resposta: o código do programa é apresentado na Figura 5.29 e o resultado da execução desse código nas Figuras 5.30, 5.31 e 5.32.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de variáveis:
6     int band,desc;
7     float kms,valor_corrida,min=3.5;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12    //Início da lógica de programação
13    printf("Digite o numero da bandeira (1 ou 2): ");
14    scanf("%d",&band);
15    printf("Digite a kilometragem rodada: ");
16    scanf("%f",&kms);
17    //Verificação do desconto:
18    printf("O taxi oferece desconto de 30% ? (1 - SIM ou 2 - NAO)\n");
19    scanf("%d",&desc);
20    //Definição dos preços/km dependendo da bandeira:
21    if(band == 1)
22        valor_corrida = 1.8*kms; //Calculo do valor da corrida
23    else if(band == 2)
24        valor_corrida = 2.3*kms; //Calculo do valor da corrida
25    if(desc == 1)
26        valor_corrida = valor_corrida*0.7;
27    if(valor_corrida<min)
28        valor_corrida=min;
29    //Resultados:
30    printf("O valor da corrida e: R$ %.2f\n", valor_corrida);
31    getch(); //Função que aguarda o recebimento de um caractere via teclado
32 }

```

Figura 5.29 – Código do programa em C: *respexe510.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Digite o numero da bandeira (1 ou 2): 2
Digite a kilometragem rodada: 32
O taxi oferece desconto de 30% ? (1 - SIM ou 2 - NAO)
1
O valor da corrida e: R$ 51.52

```

Figura 5.30 – Resultado da execução do código do programa em C: *respexe510.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira 2, e a quantidade de quilômetros rodados foi 32, com o desconto de 30%

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Digite o numero da bandeira <1 ou 2>: 2
Digite a kilometragem rodada: 32
O taxi oferece desconto de 30% ? <1 - SIM ou 2 - NAO>
2
O valor da corrida e: R$ 73.60
  
```

Figura 5.31 – Resultado do programa em C: *respexe510.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira 2, e a quantidade de quilômetros rodados foi 32, sem o desconto de 30%

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Digite o numero da bandeira <1 ou 2>: 2
Digite a kilometragem rodada: 32
O taxi oferece desconto de 30% ? <1 - SIM ou 2 - NAO>
2
O valor da corrida e: R$ 6.90
  
```

Figura 5.32 – Resultado do programa em C: *respexe510.cpp*, que ilustra um exemplo em que a corrida do taxi possui bandeira 2, e a quantidade de quilômetros rodados foi 32, com o desconto de 30%

Fonte: Elaboração própria

5.11 Dados três números, elabore um programa em C para verificar se eles formam um triângulo, caso verdadeiro calcule a sua área.

Três números só podem ser comprimento dos lados de um triângulo, se cada um deles for menor que a soma dos outros dois.

$$A < (B+C) \text{ e } B < (A+C) \text{ e } C < (A+B)$$

Caso contrário, A, B, e C não formam os lados de um triângulo.

Resposta: o código do programa é apresentado na Figura 5.33 e o resultado da execução desse código nas Figuras 5.34 e 5.35.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main () //Programa principal
5     //Declaração de Variáveis:
6     int num1, num2, num3;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Você deverá digitar três números inteiros:\n");
13    printf("Digite o primeiro número: ");
14    scanf("%d", &num1);
15    printf("Digite o segundo número: ");
16    scanf("%d", &num2);
17    printf("Digite o terceiro número: ");
18    scanf("%d", &num3);
19    //Desigualdade triangular: requisitos para que três números formem um triângulo
20    if (num1<(num2+num3))
21    {
22        if (num2<(num1+num3))
23        {
24            if (num3<(num1+num2))
25                printf("\nOs números que você digitou formam um triângulo.");
26            else
27                printf("\nOs números que você digitou não formam um triângulo.");
28        }
29        else
30            printf("\nOs números que você digitou não formam um triângulo.");
31    }
32    else
33        printf("\nOs números que você digitou não formam um triângulo.");
34    getch(); //Função que aguarda o recebimento de um caractere via teclado
35 }
```

Figura 5.33 – Código do programa em C: *respexe511.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Você deverá digitar três números inteiros:
Digite o primeiro número: 19
Digite o segundo número: 15
Digite o terceiro número: 3

Os números que você digitou não formam um triângulo.

```

Figura 5.34 – Resultado do programa em C: *respexe511.cpp*, que ilustra um exemplo em que os três números não formam um triângulo

Fonte: Elaboração própria

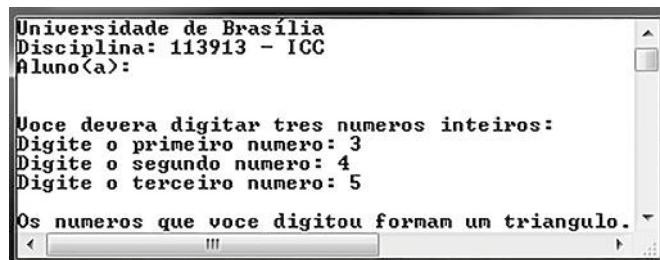


Figura 5.35 – Resultado do programa em C: *respexe511.cpp*, que ilustra um exemplo em que os três números formam um triângulo

Fonte: Elaboração própria

- 5.12 Faça um programa em C que leia três números diferentes (assuma que o sejam) e os imprima na tela em ordem crescente.

Resposta: o código do programa é apresentado na Figura 5.36 e o resultado da execução desse código nas Figuras 5.37 e 5.38.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h>/Biblioteca padrão para uso de funções do sistema
4 int main ()//Programa principal
5 {
6     //Declaração de Variáveis:
7     int num1, num2, num3;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12    //Início da lógica de programação
13    printf("Você deverá digitar três números inteiros diferentes:\n");
14    printf("Digite o primeiro numero: ");
15    scanf("%d", &num1);
16    printf("Digite o segundo numero: ");
17    scanf("%d", &num2);
18    printf("Digite o terceiro numero: ");
19    scanf("%d", &num3);
20    //Condições e Resultados
21    if (num1<num2 && num1<num3)
22    {
23        if (num2<num3)
24            printf("\nNa ordem crescente os números são: %d, %d, %d\n\n", num1, num2, num3);
25        else
26            printf("\nNa ordem crescente os números são: %d, %d, %d\n\n", num1, num3, num2);
27    }
28    else if (num2<num3)
29    {
30        if (num1<num3)
31            printf("\nNa ordem crescente os números são: %d, %d, %d\n\n", num2, num1, num3);
32        else
33            printf("\n Na ordem crescente os números são: %d, %d, %d\n\n", num2, num3, num1);
34    }
35 }
  
```

```

36     if (num1<num2)
37         printf("\n Na ordem crescente os numeros sao: %d, %d, %d\n\n", num3, num1, num2);
38     else
39         printf("\n Na ordem crescente os numeros sao: %d, %d, %d\n\n", num3, num2, num1);
40     }
41     getch(); //Função que aguarda o recebimento de um caractere via teclado
42 }
```

Figura 5.36 – Código do programa em C: *respexe512.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Voce devera digitar tres numeros inteiros diferentes:
Digite o primeiro numero: 93
Digite o segundo numero: 76
Digite o terceiro numero: 13

Na ordem crescente os numeros sao: 13, 76, 93

```

Figura 5.37 – Resultado do programa em C: *respexe512.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Voce devera digitar tres numeros inteiros diferentes:
Digite o primeiro numero: -23
Digite o segundo numero: 13
Digite o terceiro numero: 0

Na ordem crescente os numeros sao: -23, 0, 13

```

Figura 5.38 – Resultado do programa em C: *respexe512.cpp*

Fonte: Elaboração própria

- 5.13 Elabore um programa em C que leia dois números inteiros e apresente o resultado da diferença do maior número pelo menor deles.

Resposta: o código do programa é apresentado na Figura 5.39 e o resultado da execução desse código nas Figuras 5.40.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h>/Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de Variáveis:
6     int num1, num2;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Você deverá digitar dois números inteiros diferentes:\n");
13    printf("Digite o primeiro número: ");
14    scanf("%d", &num1);
15    printf("Digite o segundo número: ");
16    scanf("%d", &num2);
17    //Condições para apresentação de resultados como solicitado
18    if (num1>num2)
19        printf("\nA diferença do maior pelo menor número é: %d.\n\n", num1-num2);
20    else if(num2>num1)
21        printf("\nA diferença do maior pelo menor número é: %d.\n\n", num2-num1);
22    else
23        printf("\nOs números são iguais, a diferença entre eles é 0");
24    getch(); //Função que aguarda o recebimento de um caractere via teclado
25 }

```

Figura 5.39 – Código do programa em C: *respexe513.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Você deverá digitar dois números inteiros diferentes:
Digite o primeiro número: 576
Digite o segundo número: 63

A diferença do maior pelo menor número é: 513.

```

Figura 5.40 – Resultado do programa em C: *respexe513.cpp*

Fonte: Elaboração própria

- 5.14 Desenvolva um programa em C que leia um número inteiro, positivo ou negativo, e apresente o valor do módulo desse número.

Resposta: o código do programa é apresentado na Figura 5.41 e o resultado da execução desse código nas Figuras 5.42 e 5.43.

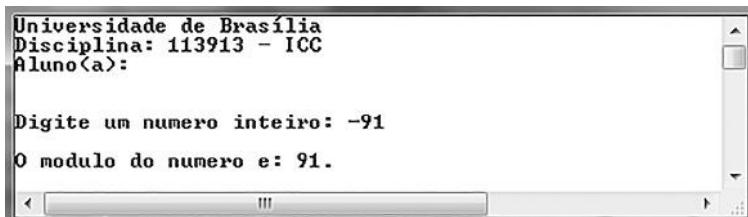
```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de Variáveis:
6     int num;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Digite um numero inteiro: ");
13    scanf("%d", &num);
14    //Condições do módulo do número
15    if (num>=0)
16        printf("\nO modulo do numero e: %d.", num);
17    else if (num<0)
18        printf("\nO modulo do numero e: %d.", -num);
19    getch(); //Função que aguarda o recebimento de um caractere via teclado
20 }

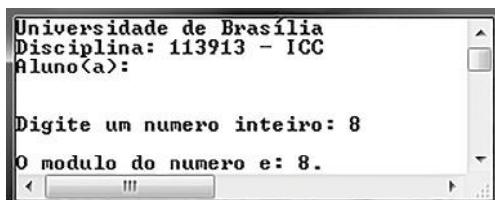
```

Figura 5.41 – Código do programa em C: *respexe514.cpp*

Fonte: Elaboração própria

Figura 5.42 – Resultado do programa em C: *respexe514.cpp*, que ilustra um exemplo em que o número digitado é negativo

Fonte: Elaboração própria

Figura 5.43 – Resultado do programa em C: *respexe514.cpp*, que ilustra um exemplo em que o número digitado é positivo

Fonte: Elaboração própria

- 5.15 Faça um programa em C que leia dois números inteiros e apresente a soma do módulo desses dois números.

Resposta: o código do programa é apresentado na Figura 5.44 e o resultado da execução desse código nas Figuras 5.45 e 5.46.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     //Declaração de Variáveis:
6     int num1, num2;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início da lógica de programação
12    printf("Você deverá digitar dois números inteiros diferentes:\n");
13    printf("Digite o primeiro número: ");
14    scanf("%d", &num1);
15    printf("Digite o segundo número: ");
16    scanf("%d", &num2);
17    //Condições do módulo para receber resultado para num1
18    if (num1<0)
19        num1=-num1;
20    //Condições do módulo para receber resultado para num2
21    if (num2<0)
22        num2=-num2;
23    printf("\nA soma dos modulos é: %d.", num1+num2);
24    getch(); //Função que aguarda o recebimento de um caractere via teclado
25 }
```

Figura 5.44 – Código do programa em C: *respexe515.cpp*

Fonte: Elaboração própria

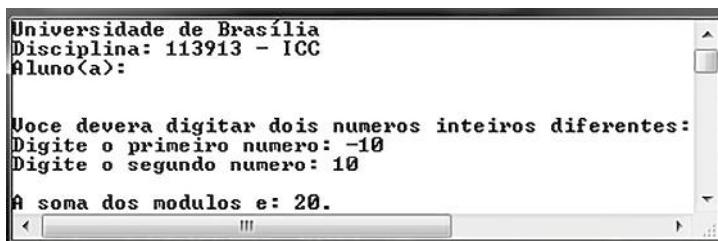


Figura 5.45 – Resultado da execução do código do programa em C: *respexe515.cpp*, que ilustra um exemplo em que o número digitado é negativo e o outro é positivo

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Você deverá digitar dois números inteiros diferentes:
Digite o primeiro número: -23
Digite o segundo número: -10
A soma dos módulos é: 33.
```

Figura 5.46 – Resultado da execução do código do programa em C: *respexe515.cpp*, que ilustra um exemplo em que os dois números digitados são negativos

Fonte: Elaboração própria

- 5.16 Faça um programa em C, que a uma vez que se forneça um número inteiro de 1 a 12, apresente o nome do mês correspondente por extenso ou uma mensagem de inválido, caso o usuário digite um número fora do estabelecido.

Resposta: o código do programa é apresentado na Figura 5.47 e o resultado da execução desse código nas Figuras 5.48, 5.49, 5.50 e 5.51.

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  int main (){//Programa principal
5      //Declaração de Variáveis:
6      int num;
7      //Cabeçalho
8      printf("Universidade de Brasília\n", 161);
9      printf("Disciplina: 113913 - ICC \n");
10     printf("Aluno(a):\n\n");
11     //Início da lógica de programação
12     printf("Digite um numero inteiro de 1 a 12 correspondente ao mes desejado: ");
13     scanf("%d", &num);
14     //Condições para verificação do mês e apresentação de resultados:
15     if (num==1)
16         printf("\nO numero corresponde ao mes de Janeiro.\n");
17     else if (num==2)
18         printf("\nO numero corresponde ao mes de Fevereiro.\n");
19     else if (num==3)
20         printf("\nO numero corresponde ao mes de Março.\n");
21     else if (num==4)
22         printf("\nO numero corresponde ao mes de Abril.\n");
23     else if (num==5)
24         printf("\nO numero corresponde ao mes de Maio.\n");
25     else if (num==6)
26         printf("\nO numero corresponde ao mes de Junho.\n");
27     else if (num==7)
28         printf("\nO numero corresponde ao mes de Julho.\n");
29     else if (num==8)
30         printf("\nO numero corresponde ao mes de Agosto.\n");
31     else if (num==9)
32         printf("\nO numero corresponde ao mes de Setembro.\n");
33     else if (num==10)
34         printf("\nO numero corresponde ao mes de Outubro.\n");
35     else if (num==11)
36         printf("\nO numero corresponde ao mes de Novembro.\n");
37     else if (num==12)
38         printf("\nO numero corresponde ao mes de Dezembro.\n");
39     else
40         printf("\nO numero não corresponde a nenhum mes do ano.\n");
41     getch(); //Função que aguarda o recebimento de um caractere via teclado
42 }

```

Figura 5.47 – Código do programa em C: *respexe516.cpp*

Fonte: Elaboração própria

```

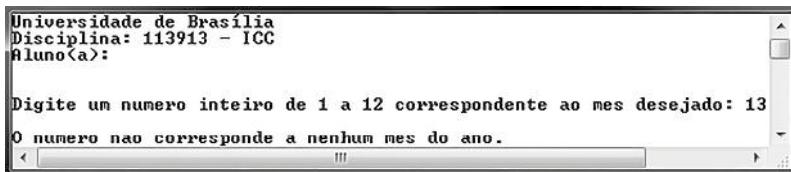
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite um numero inteiro de 1 a 12 correspondente ao mes desejado: 7
O numero corresponde ao mes de Julho.

```

Figura 5.48 – Resultado da execução do código do programa em C: *respexe516.cpp*, que ilustra um exemplo do resultado ser o mês de julho

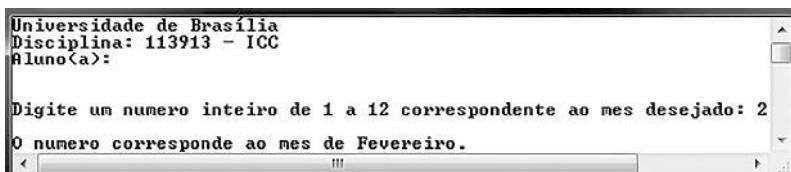
Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite um numero inteiro de 1 a 12 correspondente ao mes desejado: 13  
O numero nao corresponde a nenhum mes do ano.
```

Figura 5.49 – Resultado da execução do código do programa em C: *respexe516.cpp*, que ilustra um exemplo do resultado não corresponder a nenhum mês do ano

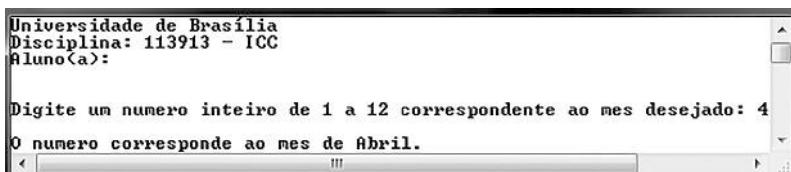
Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite um numero inteiro de 1 a 12 correspondente ao mes desejado: 2  
O numero corresponde ao mes de Fevereiro.
```

Figura 5.50 – Resultado da execução do código do programa em C: *respexe516.cpp*, que ilustra um exemplo do resultado ser o mês de fevereiro

Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite um numero inteiro de 1 a 12 correspondente ao mes desejado: 4  
O numero corresponde ao mes de Abril.
```

Figura 5.51 – Resultado da execução do código do programa em C: *respexe516.cpp*, que ilustra um exemplo do resultado ser o mês de abril

Fonte: Elaboração própria

Capítulo 6

Estrutura de Repetição Contada

Estrutura de Repetição Contada

6.1 Dado um número, crie um programa C que escreva todos os números ímpares menores e/ou iguais a esse número e maiores ou igual a um. Assuma que o número informado é positivo.

Resposta: o código do programa é apresentado na Figura 6.1 e o resultado da execução desse código na Figura 6.2.

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  int main (){//Programa principal
5      // Declaração de variáveis:
6      int num, cont;
7      //Cabeçalho
8      printf("Universidade de Brasília\n", 161);
9      printf("Disciplina: 113913 - ICC \n");
10     printf("Aluno(a):\n\n");
11     //Início do programa
12     printf("Digite um numero inteiro positivo: ");
13     scanf("%d", &num);
14     // Apresentação de resultados
15     printf("\nOs numeros impares menores que %d sao:\n", num);
16     // (for) para passar por todos os números entre 'num' e 1
17     if(num%2==0)
18         num=num-1;
19     for(cont=num; cont>=1; cont=cont-2)
20         printf("%d ", cont);
21     getch();// Função que aguarda o recebimento de um caractere via teclado
22 }
```

Figura 6.1 – Código do programa em C: *respexe61.cpp*

Fonte: Elaboração própria

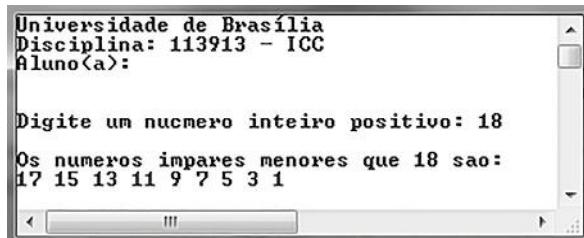


Figura 6.2 – Resultado da execução do código do programa em C: respexe61.cpp

Fonte: Elaboração própria

6.2 Dado um conjunto de N números, faça um programa C que calcule e mostre a média aritmética dos elementos desse conjunto.

Resposta: o código do programa é apresentado na Figura 6.3 e o resultado da execução desse código na Figura 6.4.

```

1 #include<stdio.h> //declarando biblioteca de entrada/saída
2 #include<conio.h> //declarando biblioteca para uso de funções auxiliares
3 #include<stdlib.h> //declarando biblioteca para uso de funções auxiliares
4 int main()
5 {
6     int n, cont; // Declaração de variáveis para receber a quantidade de números e manipular o 'for'
7     float num, soma=0, media; // Declaração de variáveis para receber cada número,
8     //para acumular soma, e realizar a média
9     system("color f0");
10    // Cabeçalho
11    printf("Universidade de Brasília\n", 161);
12    printf("Disciplina: 113913 - Introdução à Ciência da Computação \n", 135, 133, 135);
13    printf("Aluno:\n\n");
14    // Mensagem ao usuário para que esse insira o dado via teclado
15    printf("De quantos números você deseja calcular a média: ");
16    scanf("%d", &n);
17
18    if (n<0)//Validação simples de dados
19    {
20        printf("\nDados incorretos.\n O número deve ser positivo.\n");
21        scanf("%d", &n);
22    }
23    // Laço para receber os números quantas vezes foi solicitado
24    for (cont=1; cont<=n; cont++)
25    {
26        printf("Digite um dos números: ");
27        scanf("%f", &num);
28        soma=soma+num; // Acumular a soma
29    }
30    media=soma/n; // Calculo da média e apresentação dos resultados
31    printf("\n\nA média dos números fornecidos é: %.1f\n\n", media);
32    getch();
}

```

Figura 6.3 – Código do programa em C: respexe62.cpp

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - Introdução à Ciencia da Computação
Aluno:

De quantos numeros voce deseja calcular a media: 3
Digite um dos numero: 9
Digite um dos numero: 3
Digite um dos numero: 8

A media dos numeros fornecidos e: 6.7
  
```

Figura 6.4 – Resultado da execução do código do programa em C: *respexe62.cpp*

Fonte: Elaboração própria

6.3 A conversão de graus Fahrenheit para Celsius é obtida por $T_c = \{(T_f - 32) \times (5/9)\}$, em que T_c é a temperatura em Celsius e T_f em Fahrenheit. Faça um programa C que calcule e que imprima uma tabela de graus Fahrenheit e graus Celsius, cujos graus variem de 50 a 65, de 1 em 1.

Resposta: o código do programa é apresentado na Figura 6.5 e o resultado da execução desse código na Figura 6.6.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     float Tc;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno (a):\n\n");
11    //Início do programa
12    //Laço apenas com uma variável pois os valores limites já foram definidos
13    for (Tc=50; Tc<=65; Tc++)
14        printf(" %.1f °C = %.1f °F \n", Tc, 248, ((Tc*9.0)/5.0)+32.0), 248);
15    getch();// Função que aguarda o recebimento de um caractere via teclado
16 }
  
```

Figura 6.5 – Código do programa em C: *respexe63.cpp*

Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

50 °C = 122.0 °F
51 °C = 123.8 °F
52 °C = 125.6 °F
53 °C = 127.4 °F
54 °C = 129.2 °F
55 °C = 131.0 °F
56 °C = 132.8 °F
57 °C = 134.6 °F
58 °C = 136.4 °F
59 °C = 138.2 °F
60 °C = 140.0 °F
61 °C = 141.8 °F
62 °C = 143.6 °F
63 °C = 145.4 °F
64 °C = 147.2 °F
65 °C = 149.0 °F
```

Figura 6.6 – Resultado da execução do código do programa em C: *respexe63.cpp*

Fonte: Elaboração própria

- 6.4 Faça um programa C que leia dez números que representem as notas de dez alunos, e que apresente:
- a soma dos números;
 - a média dos números;
 - o maior número;
 - o menor número.

Assuma que as notas são informadas corretamente no intervalo de 1 a 10.

Resposta: o código do programa é apresentado na Figura 6.7 e o resultado da execução desse código nas Figuras 6.8 e 6.9.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int cont;
7     float nota, soma=0, maior=0, menor=10;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12    //Início do programa
13    // Receber cada nota realizando as operações e comparações solicitadas
14    for (cont=1; cont<=10; cont++)
15    {
16        printf("Digite a %d.a nota: ", cont);
17        scanf("%f", &nota);
18        // Validação simples de nota
19        if (nota<0 || nota>10)
20        {
21            printf("\nDados incorretos. A nota deve estar entre 0 e 10: ");
22            scanf("%f", &nota);
23        }
24        soma=soma+nota;// Acumulo da soma
25        if (nota>maior) // Comparação para obter a maior nota
26        {
27            maior=nota;
28        }
29        if (nota<menor)// Comparação para obter a menor nota
30        {
31            menor=nota;
32        }
33        // Apresentação dos resultados
34        printf("\nA soma das notas é: %.1f", soma);
35        printf("\nA média das notas fornecidas é: %.1f", soma/10);
36        printf("\nA maior nota é: %.1f", maior);
37        printf("\nA menor nota é: %.1f", menor);
38        getch(); // Função que aguarda o recebimento de um caractere via teclado

```

Figura 6.7 – Código do programa em C: *respexe64.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite a 1.a nota: 10
Digite a 2.a nota: 9.5
Digite a 3.a nota: 9.5
Digite a 4.a nota: 6
Digite a 5.a nota: 7.5
Digite a 6.a nota: 4
Digite a 7.a nota: 2
Digite a 8.a nota: 10
Digite a 9.a nota: 7.5
Digite a 10.a nota: 8

A soma das notas e: 74.0
A media das notas fornecidas e: 7.4
A maior nota e: 10.0
A menor nota e: 2.0
```

Figura 6.8 – Resultado da execução do código do programa em C: *respexe64.cpp*, que ilustra um exemplo com a maior nota igual a 10 e a menor igual 2

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite a 1.a nota: 9.5
Digite a 2.a nota: 10
Digite a 3.a nota: 3.5
Digite a 4.a nota: 8
Digite a 5.a nota: 8.5
Digite a 6.a nota: 8
Digite a 7.a nota: 7.5
Digite a 8.a nota: 4
Digite a 9.a nota: 5
Digite a 10.a nota: 9

A soma das notas e: 73.0
A media das notas fornecidas e: 7.3
A maior nota e: 10.0
A menor nota e: 3.5
```

Figura 6.9 – Resultado da execução do código do programa em C: *respexe64.cpp*, que ilustra um exemplo com a maior nota igual a 10 e a menor igual 3,5

Fonte: Elaboração própria

- 6.5 Desenvolva um programa, maioridade.cpp, que solicite ao usuário a idade de três pessoas e apresente na tela a maior e a menor idade.

Resposta: o código do programa é apresentado na Figura 6.10 e o resultado da execução desse código nas Figuras 6.11 e 6.12.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main ()//Programa principal
5     // Declaração de variáveis:
6     int idade, cont, maior=0, menor=1000;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início do programa
12    for(cont=1; cont<=3; cont++)
13    {
14        printf("Digite a %d.a idade: ", cont);
15        scanf("%d", &idade);
16        if(idade<0)
17        {
18            printf("Digite a %d.a idade, deve ser positivo: ", cont);
19            scanf("%d", &idade);
20        }
21        if(idade>maior)
22            maior=idade;
23        if(idade<menor)
24            menor=idade;
25    }
26    printf("\nA maior idade é: %d", maior);
27    printf("\nA menor idade é: %d", menor);
28    getch(); // Função que aguarda o recebimento de um caractere via teclado
29 }
```

Figura 6.10 – Código do programa em C: *respexe65.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite a 1.a idade: 15
Digite a 2.a idade: 12
Digite a 3.a idade: 56  
  
A maior idade é: 56
A menor idade é: 12
```

Figura 6.11 – Resultado da execução do código do programa em C: *respexe65.cpp*, que ilustra um exemplo com idades de 12, 15 e 56 anos

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite a 1.a idade: 89
Digite a 2.a idade: 65
Digite a 3.a idade: 70  
  
A maior idade é: 89
A menor idade é: 65
```

Figura 6.12 – Resultado da execução do código do programa em C: *respexe65.cpp*, que ilustra um exemplo com idades de 65, 70 e 89 anos

Fonte: Elaboração própria

6.6 Desenvolva um programa, *tabuada.cpp*, que apresente a tabuada de 0 até 9 para o usuário. Entre cada tabuada, coloque uma pausa para o usuário conseguir lê-la.

Resposta: o código do programa é apresentado na Figura 6.13 e a execução desse código na Figura 6.14.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int cont, cont2;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início do programa
12    // Laço de repetição para a tabuada
13    for(cont=1; cont<9; cont++){
14        for(cont2=1; cont2<=10; cont2++){
15            printf("%d x %d = %d\n", cont, cont2, cont*cont2);
16            printf("\n");
17            getch(); //Aguardar comando para continuar
18        }
19    }
20    getch(); // Função que aguarda o recebimento de um caractere via teclado
}

```

Figura 6.13 – Código do programa em C: *respexe66.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10

```

Figura 6.14 – Resultado da execução do código do programa em C: *respexe66.cpp*

Fonte: Elaboração própria

- 6.7 Desenvolva um programa, *tabinter.cpp*, para calcular a tabuada de um intervalo de números que será informado pelo usuário.

Resposta: o código do programa é apresentado na Figura 6.15 e a execução desse código na Figura 6.16.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int cont, cont2, num1, num2;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início do programa
12    printf("Digite o primeiro limite do intervalo: ");
13    scanf("%d", &num1);
14    printf("Digite o segundo limite do intervalo: ");
15    scanf("%d", &num2);
16    if(num2<=num1)
17    {
18        printf("\tO segundo limite deve ser maior que o primeiro\n");
19        printf("Digite o primeiro limite do intervalo: ");
20        scanf("%d", &num1);
21        printf("Digite o segundo limite do intervalo: ");
22        scanf("%d", &num2);
23    }
24    // Laço de repetição para a tabuada
25    for(cont=num1; cont<=num2; cont++)
26    {
27        for(cont2=1; cont2<=10; cont2++)
28            printf("%d x %d = %d\n", cont, cont2, cont*cont2);
29        printf("\n");
30        getch(); //Aguardar comando para continuar
31    }
32    getch(); // Função que aguarda o recebimento de um caractere via teclado.
33 }
```

Figura 6.15 – Código do programa em C: *respexe67.cpp*

Fonte: Elaboração própria

```
Digite o primeiro limite do intervalo: 3
Digite o segundo limite do intervalo: 5
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30

4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
```

Figura 6.16 – Resultado da execução do código do programa em C: *respexe67.cpp*

Fonte: Elaboração própria

Capítulo 7

**Estrutura de Repetição
Condicional com Teste no Início**

Condisional com Teste no Início

7.1 Dado um número, crie um programa C que escreva todos os números ímpares menores que o número informado e maiores do que um. Assuma que o número informado é positivo.

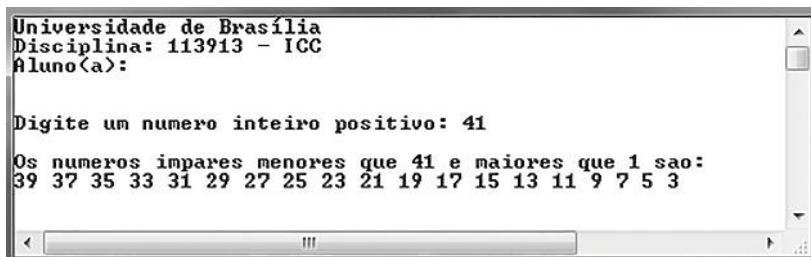
Resposta: o código do programa é apresentado na Figura 7.1 e o resultado da execução desse código na Figura 7.2.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int num;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início do programa
12    printf("Digite um numero inteiro positivo: ");
13    scanf("%d", &num);
14    //Enquanto num for negativo, será pedido novamente
15    while(num<0)
16    {
17        printf("\nErro!Digite um numero inteiro positivo: ");
18        scanf("%d", &num);
19    }
20
21    printf("\nOs numeros ímpares menores que %d e maiores que 1 são:\n", num);
22    /* Condicionando num para que seja menor que este e que seja ímpar*/
23    if (num%2==0)
24    {
25        num--;
26    }
27    /* Como já foi condicionado num como numero ímpar e como sendo menor que o fornecido
28    -basta tirar dois e imprimi-los/
29    while (num>1)
30    {
31        printf("%d ", num);
32        num=num-2;
33    }
34    getch();// Função que aguarda o recebimento de um caractere via teclado
35 }
```

Figura 7.1 – Código do programa em C: *respexe71.cpp*

Fonte: Elaboração própria



The screenshot shows a terminal window with the following text:

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite um numero inteiro positivo: 41
Os numeros impares menores que 41 e maiores que 1 sao:
39 37 35 33 31 29 27 25 23 21 19 17 15 13 11 9 7 5 3
```

Figura 7.2 – Resultado da execução do código do programa em C: *respexe71.cpp*

Fonte: Elaboração própria

- 7.2 Faça um programa que exiba a tabuada dos números de 10 a 20. Por exemplo:

10X0=0

10X1=10

....

10X10=100

até chegar à tabuada de 20,

20X0=0

20X1=20

....

20X10=200

Resposta: o código do programa é apresentado na Figura 7.3 e o resultado da execução desse código na Figura 7.4.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int cont, auxcont=10;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início do programa
12    // Ciclo de repetição externo para incrementar auxcont "tabuada do numero auxcont"
13    while (auxcont<=20)
14    {
15        cont=0; //Importante zerar cont pois será usada novamente
16        //Laço interno para calculo e impressão de resultados
17        while (cont<10)
18        {
19            printf("\t%d x %d = %d\n", auxcont, cont, auxcont*cont);
20            cont++;
21        }
22        printf("\n\n");
23        auxcont++;
24    }
25    getch();// Função que aguarda o recebimento de um caractere via teclado
26 }

```

Figura 7.3 – Código do programa em C: *respexe72.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

      10 x 0 = 0
      10 x 1 = 10
      10 x 2 = 20
      10 x 3 = 30
      10 x 4 = 40
      10 x 5 = 50
      10 x 6 = 60
      10 x 7 = 70
      10 x 8 = 80
      10 x 9 = 90
      10 x 10 = 100

      11 x 0 = 0
      11 x 1 = 11
      11 x 2 = 22
      11 x 3 = 33
      11 x 4 = 44
      11 x 5 = 55
      11 x 6 = 66
      11 x 7 = 77
      11 x 8 = 88
      11 x 9 = 99
      11 x 10 = 110

      12 x 0 = 0
      12 x 1 = 12
  
```

Figura 7.4 – Resultado da execução do código do programa em C: *respexe72.cpp*

Fonte: Elaboração própria

7.3 Um funcionário de uma empresa recebe aumento salarial anualmente. Sabe-se que:

- esse funcionário foi contratado em 1995, com salário inicial de R\$ 1.000,00;
- em 1996 recebeu aumento de 1,5% sobre seu salário inicial;
- a partir de 1997 (inclusive), os aumentos salariais sempre corresponderam ao dobro do percentual do ano anterior.

Faça um programa que determine o salário desse funcionário até o ano 2000 (caso você tenha uma excelente máquina, poderá realizar a operação até o ano final desejado).

Resposta: o código do programa é apresentado na Figura 7.5 e a execução desse código nas Figuras 7.6 e 7.7.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main ()//Programa principal
5     // Declaração de variáveis:
6     int cont, ano;
7     double salario=1015.0, porcentagem=0.015;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 16);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12    //Início do programa
13    printf("Digite o ano, a partir de 1996 que deseja saber o valor do salário: ");
14    scanf("%d", &ano);
15    //Validação em relação ao ano mínimo e a um ano que se queira como limite, ex:2020
16    while((ano<1996)|| (ano>2000))
17    {
18        printf("Erro! Digite o ano, entre 1996 e 2000: ");
19        scanf("%d", &ano);
20    }
21    // diferença entre o ano desejado e o ano inicial para manipulação do laço de repetição
22    cont=ano-1996;
23    // Laço para relacionar o dobro da porcentagem ao ano seguinte e imprimi-lo
24    while (cont>=1)
25    {
26        porcentagem=porcentagem*2;
27        salario+=(salario*porcentagem);
28        //Impressão dos salários entre o ano de 1997 e o ano desejado
29        printf("\nR$%.2f ", (ano-cont)+1, salario);
30        cont--;
31    }
32    //Apresentação dos resultados
33    printf("\nO salário do ano de %d sera: R$%.2f", ano, salario);
34    getch(); // Função que guarda o recebimento de um caractere via teclado
35 }
```

Figura 7.5 – Código do programa em C: *respexe73.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o ano, a partir de 1996 que deseja saber o valor do salario: 1999
1997 = R$1045.45
1998 = R$1108.18
1999 = R$1241.16
O salario do ano de 1999 sera: R$1241.16
  
```

Figura 7.6 – Resultado da execução do código do programa em C: *respexe73.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o ano, a partir de 1996 que deseja saber o valor do salario: 2000
1997 = R$1045.45
1998 = R$1108.18
1999 = R$1241.16
2000 = R$1539.04
O salario do ano de 2000 sera: R$1539.04
  
```

Figura 7.7 – Resultado da execução do código programa em C: *respexe73.cpp*

Fonte: Elaboração própria

- 7.4 Faça um programa que leia dez conjuntos de dois valores, o primeiro representando o número do aluno e o segundo representando a sua altura em centímetros. Encontre o aluno mais alto e o mais baixo. Apresente o número do aluno mais alto e o do aluno mais baixo, cada um com sua altura.

Resposta: o código do programa é apresentado na Figura 7.8 e a execução desse código na Figura 7.9.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main ()//Programa principal
5     // Declaração de variáveis:
6     int cont=1, numero, maior_num, menor_num;
7     float altura, menor_altura=3.00, maior_altura=0.0;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12    //Início do programa
13    // Laço para receber os dados dos dez alunos
14    while (cont<=10)
15    {
16        printf("Digite o numero do %d.o aluno: ", cont); // receber o identificar numeral
17        scanf("%d", &numero);
18        while(numero<0) //Validação de numero
19        {
20            printf("Erro!! Digite novamente o numero %d.o aluno: ", cont);
21            scanf("%d", &numero);
22        }
23        printf("Digite a altura do %d.o aluno: ", cont);
24        scanf("%f", &altura);
25        while((altura<0)|| (altura>3)) //Validação da altura
26        {
27            printf("Erro!! Digite novamente a altura do %d.o aluno: ", cont);
28            scanf("%f", &altura);
29        }
30        if (altura>maior_altura)// identificar e atribuir o maior aluno.
31        {
32            maior_altura=altura;
33            maior_num=numero;
34        }
35        if (altura<menor_altura) // identificar e atribuir o menor aluno.
36        {
37            menor_altura=altura;
38            menor_num=numero;
39        }
40        cont++;
41
42     //Apresentar os resultados
43     printf("\nO aluno mais baixo é o de numero %d com %.2fm.\n", menor_num, menor_altura);
44     printf("O aluno mais alto é o de numero %d com %.2fm.", maior_num, maior_altura);
45     getch(); // Função que aguarda o recebimento de um caractere via teclado
46 }

```

Figura 7.8 – Código do programa em C: *respexe74.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o numero do 1.o aluno: 1
Digite a altura do 1.o aluno: 1.45
Digite o numero do 2.o aluno: 2
Digite a altura do 2.o aluno: 1.54
Digite o numero do 3.o aluno: 3
Digite a altura do 3.o aluno: 1.9
Digite o numero do 4.o aluno: 4
Digite a altura do 4.o aluno: 1.67
Digite o numero do 5.o aluno: 5
Digite a altura do 5.o aluno: 1.81
Digite o numero do 6.o aluno: 6
Digite a altura do 6.o aluno: 1.84
Digite o numero do 7.o aluno: 7
Digite a altura do 7.o aluno: 1.41
Digite o numero do 8.o aluno: 8
Digite a altura do 8.o aluno: 1.89
Digite o numero do 9.o aluno: 9
Digite a altura do 9.o aluno: 1.75
Digite o numero do 10.o aluno: 10
Digite a altura do 10.o aluno: 1.73

O aluno mais baixo e o de numero 7 com 1.41m.
O aluno mais alto e o de numero 3 com 1.90m.

```

Figura 7.9 – Resultado da execução do código programa em C: *respexe74.cpp*

Fonte: Elaboração própria

7.5 Faça um programa que mostre todos os números pares existentes entre 1 e 50.

Resposta: o código do programa é apresentado na Figura 7.10 e o resultado da execução desse código na Figura 7.11.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int cont=2; //Variável referência para o laço de repetição
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início do programa
12    //Laço para vai de 2 à 48 que são os extremos numeros pares entre 1 e 50
13    while (cont<=48)
14    {
15        printf("%d ", cont);
16        cont=cont+2;
17    }
18 }
19 getch()// Função que aguarda o recebimento de um caractere via teclado

```

Figura 7.10 – Código do programa em C: *respexe75.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):
2  4  6  8  10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42
44 46 48

```

Figura 7.11 – Resultado da execução do código do programa em C: *respexe75.cpp*

Fonte: Elaboração própria

7.6 Desenvolva um programa que apresente a tabuada de 0 até 9 para o intervalo de 0 até 10. Entre cada tabuada, coloque uma pausa para o usuário conseguir lê-la.

Resposta: o código do programa é apresentado na Figura 7.12 e o resultado da execução desse código na Figura 7.13.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int cont1, auxcont; //Declaração de variáveis
//Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início do programa
12    // Laco de repetição para a tabuada
13    while(cont1<=9)
14    {
15        auxcont=0;
16        while(auxcont<=10)
17        {
18            printf("%d x %d = %d\n", cont1, auxcont, cont1*auxcont);
19            auxcont++;
20        }
21        getch();
22        cont1++;
23    }
24    getch(); // Função que aguarda o recebimento de um caractere via teclado
25 }

```

Figura 7.12 – Código do programa em C: *respexe76.cpp*

Fonte: Elaboração própria

<code>1 x 0 = 0 1 x 1 = 1 1 x 2 = 2 1 x 3 = 3 1 x 4 = 4 1 x 5 = 5 1 x 6 = 6 1 x 7 = 7 1 x 8 = 8 1 x 9 = 9 1 x 10 = 10</code>	<code>5 x 0 = 0 5 x 1 = 5 5 x 2 = 10 5 x 3 = 15 5 x 4 = 20 5 x 5 = 25 5 x 6 = 30 5 x 7 = 35 5 x 8 = 40 5 x 9 = 45 5 x 10 = 50</code>	<code>9 x 0 = 0 9 x 1 = 9 9 x 2 = 18 9 x 3 = 27 9 x 4 = 36 9 x 5 = 45 9 x 6 = 54 9 x 7 = 63 9 x 8 = 72 9 x 9 = 81 9 x 10 = 90</code>
--	--	--

Figura 7.13 – Resultado da execução do código do programa em C: *respexe76.cpp*

Fonte: Elaboração própria

- 7.7 Desenvolva um programa para calcular a tabuada de um intervalo de números que será informado pelo usuário.

Resposta: o código do programa é apresentado na Figura 7.14 e o resultado da execução desse código na Figura 7.15.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main ()//Programa principal
5 // Declaração de variáveis:
6 int cont, auxcont, num1, num2;
7 //Cabeçalho
8 printf("Universidade de Brasília\n", 161);
9 printf("Disciplina: 113913 - ICC \n");
10 printf("Aluno(a):\n\n");
11 //Início do programa
12 printf("Digite o primeiro numero: ");
13 scanf("%d", &num1);
14 printf("Digite o segundo numero: ");
15 scanf("%d", &num2);
16 cont=num1;
17 // Laço de repetição para a tabuada
18 while(cont<=num2)
19 {
20     auxcont=1;
21     while(auxcont<=10)
22     {
23         printf("%d x %d = %d\n", cont, auxcont, cont*auxcont);
24         auxcont++;
25     }
26     printf("\n");
27     getch(); // Função que aguarda o recebimento de um caractere via teclado
28     cont++;
29 }
30 getch(); // Função que aguarda o recebimento de um caractere via teclado
31 }
```

Figura 7.14 – Código do programa em C: *respexe77.cpp*

Fonte: Elaboração própria

```

Digite o primeiro numero: 13
Digite o segundo numero: 16
13 x 1 = 13
13 x 2 = 26
13 x 3 = 39
13 x 4 = 52
13 x 5 = 65
13 x 6 = 78
13 x 7 = 91
13 x 8 = 104
13 x 9 = 117
13 x 10 = 130

14 x 1 = 14
14 x 2 = 28
14 x 3 = 42
14 x 4 = 56
14 x 5 = 70
  
```

Figura 7.15 – Resultado da execução do código do programa em C: *respexe77.cpp*

Fonte: Elaboração própria

- 7.8 Faça um programa que apresente a soma acumulada de todos os valores entre 1 e 100, somando os números 1 e 100 (inclusive).

Resposta: o código do programa é apresentado na Figura 7.16 e o resultado da execução desse código na Figura 7.17.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main () //Programa principal
5 {
6     // Declaração de variáveis:
7     int cont=1, soma=0;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n\n");
12    //Início do programa
13    while (cont<=100)
14    {
15        soma+=cont;
16        cont++;
17    }
18    printf("A soma acumulada de todos os valores entre 0 e 100 é = %d", soma);
19    getch(); // Função que aguarda o recebimento de um caractere via teclado
20 }
  
```

Figura 7.16 – Código do programa em C: *respexe78.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  

A soma acumulada de todos os valores entre 0 e 100 e = 5050

```

Figura 7.17 – Resultado da execução do programa em C: *respexe78.cpp*

Fonte: Elaboração própria

- 7.9 Dado um conjunto de N números, faça um programa C que calcule e mostre a média aritmética do elementos desse números.

Resposta: o código do programa é apresentado na Figura 7.18 e o resultado da execução desse código na Figura 7.19.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main ()//Programa principal
5     // Declaração de variáveis:
6     int qnt, cont;
7     float num, soma=0;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12    // Início do programa
13    printf("Digite a quantidade de numeros que voce deseja calcular a media: ");
14    scanf("%d", &qnt);
15    // While para realizar validação
16    while (qnt<0)
17    {
18        printf("ERRO! Digite novamente a quantidade de numeros para calculo da média: ");
19        scanf("%d", &qnt);
20    }
21    cont=qnt; // Atribui-se à cont o valor de n para manipulação no laço de repetição.
22    //Laço para acumulo da soma de uma quantidade de numeros
23    printf("\n");
24    while (cont>=1)
25    {
26        printf("Digite o %d.o numero: ", (qnt-cont+1));
27        scanf("%f", &num);
28        soma+=num;
29        cont--;
30    }
31    // Apresentação do Resultado
32    printf("\nA média dos numeros fornecidos é: %.1f ", (soma/qnt));
33    getch(); // Função que aguarda o recebimento de um caractere via teclado
34

```

Figura 7.18 – Código do programa em C: *respexe79.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite a quantidade de numeros que voce deseja calcular a media: 3
Digite o 1.o numero: 254
Digite o 2.o numero: 98.2
Digite o 3.o numero: 2.45
A media dos numeros fornecidos e: 118.2

```

Figura 7.19 – Resultado do programa em C: respexe79.cpp

Fonte: Elaboração própria

- 7.10 A conversão de graus Fahrenheit para Celsius é obtida por $T_c = \frac{(T_f - 32) \times 5}{9}$, em que T_c é a temperatura em graus Celsius e T_f em Fahrenheit. Faça um programa C que calcule e que imprima uma tabela de graus Fahrenheit e graus Celsius, cujos graus variem de 50 a 65, de 1 em 1.

Resposta: o código do programa é apresentado na Figura 7.20 e o resultado da execução desse código na Figura 7.21.

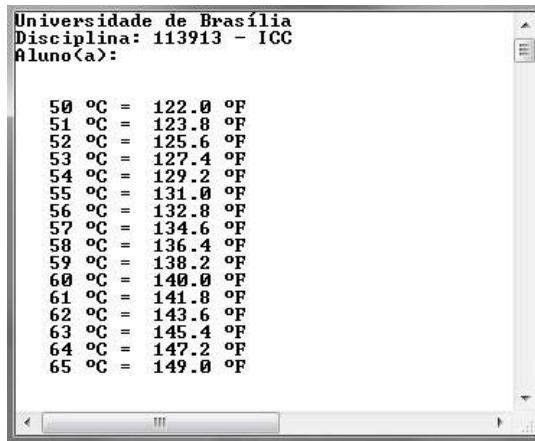
```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     float Tc=50.0;
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início do programa
12    // Laço para realização do cálculo e apresentação do resultado
13    while (Tc<=65.0)
14    {
15        printf("\t%.1f C \t=\t %.1f \tF \n", Tc, (((Tc*9.0)/5.0)+32.0));
16        Tc++;
17    }
18    getch(); // Função que aguarda o recebimento de um caractere via teclado
19 }

```

Figura 7.20 – Código do programa em C: respexe710.cpp

Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

50 °C = 122.0 °F
51 °C = 123.8 °F
52 °C = 125.6 °F
53 °C = 127.4 °F
54 °C = 129.2 °F
55 °C = 131.0 °F
56 °C = 132.8 °F
57 °C = 134.6 °F
58 °C = 136.4 °F
59 °C = 138.2 °F
60 °C = 140.0 °F
61 °C = 141.8 °F
62 °C = 143.6 °F
63 °C = 145.4 °F
64 °C = 147.2 °F
65 °C = 149.0 °F
```

Figura 7.21 – Resultado do programa em C: *respexe710.cpp*

Fonte: Elaboração própria

- 7.11 Faça um programa C que leia dez números que representam as notas de dez alunos, e obtenha:
- a soma dos números;
 - a média dos números;
 - o maior número;
 - o menor número.

Assuma que as notas são informadas corretamente no intervalo de 1 a 10.

Resposta: o código do programa é apresentado na Figura 7.22 e a execução desse código na Figura 7.23.

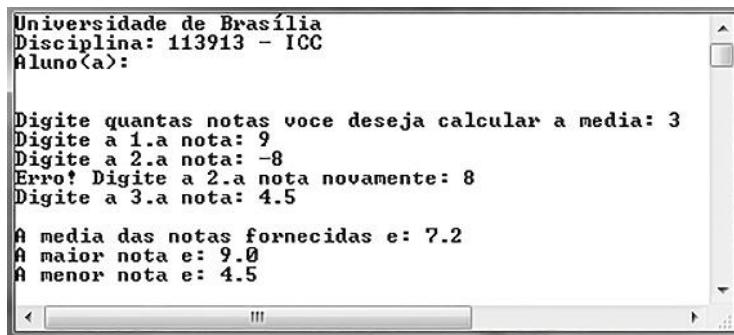
```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  int main ()//Programa principal
5      // Declaração de variáveis:
6      int qnt, cont;
7      float nota, soma=0, maior=0, menor=10;
8      //Cabeçalho
9      printf("Universidade de Brasília\n", 161);
10     printf("Disciplina: 113913 - ICC \n");
11     printf("Aluno(a):\n\n");
12     //Início do programa
13     printf("Digite quantas notas você deseja calcular a média: ");
14     scanf("%d", &qnt);
15     //Teste de validação
16     while(qnt<0)
17     {
18         printf("\nERRO! Digite novamente a quantidade de notas: ");
19         scanf("%d", &qnt);
20     }
21     cont=1; // Atribuição para manipulação dentro do laço de repetição
22     //Laço para receber todas as notas
23     while (cont<=qnt)
24     {
25         printf("Digite a %d.a nota: ", (cont));
26         scanf("%f", &nota);
27         while((nota<0) || (nota>10))
28         {
29             printf("Erro! Digite a %d.a nota novamente: ", (cont));
30             scanf("%f", &nota);
31         }
32         soma=soma+nota; // acumulo das notas
33         if (nota>maior)// teste de comparação para atribuição caso verdadeiro
34             maior=nota;
35         if (nota<menor)// teste de comparação para atribuição caso verdadeiro
36             menor=nota;
37         cont++;
38     }
39     // Apresentação dos resultados
40     printf("\nA meia das notas fornecidas é: %.1f", (soma/qnt));
41     printf("\nA maior nota é: %.1f", maior);
42     printf("\nA menor nota é: %.1f", menor);
43     getch(); // Função que aguarda o recebimento de um caractere via teclado
44 }

```

Figura 7.22 – Código do programa em C: *respexe711.cpp*

Fonte: Elaboração própria



Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

```
Digite quantas notas voce deseja calcular a media: 3
Digite a 1.a nota: 9
Digite a 2.a nota: -8
Erro! Digite a 2.a nota novamente: 8
Digite a 3.a nota: 4.5

A media das notas fornecidas e: 7.2
A maior nota e: 9.0
A menor nota e: 4.5
```

Figura 7.23 – Resultado da execução do código do programa em C: *respexe711.cpp*

Fonte: Elaboração própria

Capítulo 8

**Estrutura de Repetição
Condisional com Teste no Final**

Condisional com Teste no Final

8.1 Dado um número, crie um programa C que escreva todos os números ímpares menores que esse número e maiores do que um. Assuma que o número informado é positivo.

Resposta: o código do programa é apresentado na Figura 8.1 e o resultado da execução desse código na Figura 8.2.

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  int main () //Programa principal
5  {
6      // Declaração de variáveis:
7      int num;
8      //Cabeçalho
9      printf("Universidade de Brasília\n", 161);
10     printf("Disciplina: 113913 - ICC \n");
11     printf("Aluno(a):\n\n");
12     //Início do programa
13     /*Validação baseada no enunciado. Não devem ser mostrados
14     os numero digitado e o um. Para o caso de três ou menor,
15     não seria mostrado nenhum número*/
16     do
17     {
18         printf("Digite um numero inteiro positivo maior que 3: ");
19         scanf("%d", &num);
20     while(num<=3);
21     printf("\nOs numeros impares menores que %d e maiores que 1 sao:\n", num);
22     //Condicionando o num para ser impar e menor do que o número dado.
23     if (num%2==0)
24         num--;
25     else
26         num=num-2;
27     // Laço para manipulação e decremento do número informado
28     do
29     {
30         printf("%d ", num);
31         num=num-2;
32     }
33     while (num>=3); // condição: maior que um.
34     getch(); // Função que aguarda o recebimento de um caractere via teclado
}

```

Figura 8.1 – Código do programa em C: *respexe81.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite um numero inteiro positivo maior que 3: 49
Os numeros impares menores que 49 e maiores que 1 sao:
47 45 43 41 39 37 35 33 31 29 27 25 23 21 19 17 15 13 11 9 7 5 3

```

Figura 8.2 – Resultado da execução do código do programa em C: *respexe81.cpp*

Fonte: Elaboração própria

8.2 Dado um conjunto de N números, faça um programa C que calcule e que mostre a média aritmética dos elementos do conjunto.

Resposta: o código do programa é apresentado na Figura 8.3 e o resultado da execução desse código na Figura 8.4.

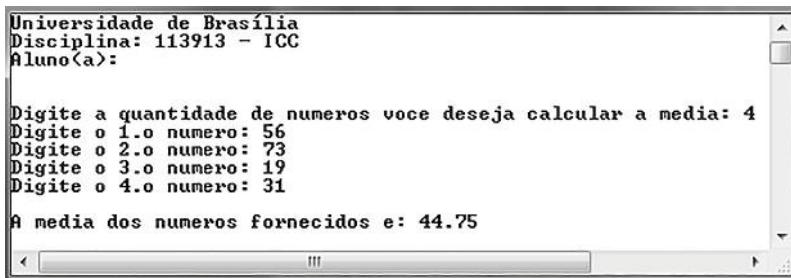
```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main ()//Programa principal
5     // Declaração de variáveis:
6     int qnt, cont;
7     float num, soma=0;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12    //Início do programa
13    printf("Digite a quantidade de números voce deseja calcular a media: ");
14    scanf("%d", &qnt);
15    while(qnt<=0)
16    {
17        printf("Erro! Digite a quantidade de números voce deseja calcular a media: ");
18        scanf("%d", &qnt);
19    }
20    cont=1;// atribuição para manipulação a partir do número obtido
21    do
22    {
23        printf("Digite o %d.o numero: ", cont);
24        scanf("%f", &num);
25        soma+=num;
26        cont++;
27    }while (cont<=qnt);
28    printf("\nA media dos numeros fornecidos e: %.2f \n\n", (soma/qnt));
29    getch(); // Função que aguarda o recebimento de um caractere via teclado
30
}

```

Figura 8.3 – Código do programa em C: *respexe82.cpp*

Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite a quantidade de numeros voce deseja calcular a media: 4
Digite o 1.o numero: 56
Digite o 2.o numero: 73
Digite o 3.o numero: 19
Digite o 4.o numero: 31

A media dos numeros fornecidos e: 44.75
```

Figura 8.4 – Resultado da execução do código do programa em C: *respexe82.cpp*

Fonte: Elaboração própria

8.3 Faça um programa C que leia dez números que representam as notas de dez alunos de uma disciplina. As notas variam de zero até dez (0 a 10). O programa deve validar a entrada de dados e obter:

- a) a soma das notas;
- b) a média das notas;
- c) a maior nota;
- d) a menor nota.

Assuma que as notas são informadas corretamente no intervalo de 1 a 10.

Resposta: o código do programa é apresentado na Figura 8.5 e o resultado da execução desse código na Figura 8.6.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int cont=1; // Contador
7     float nota, soma=0, maior=0, menor=10;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n\n");
12    //Início do programa
13    printf("Digite 10 notas para calcular a média: \n");
14    // As notas serão acumuladas e comparadas no mesmo laço instantaneamente
15    do
16    {
17        printf("Digite a %d.a nota: ", cont);
18        scanf("%f", &nota);
19        while (nota<0 || nota>10) // validação
20        {
21            printf("Dados incorretos. A nota deve estar entre 0 e 10. \nDigite novamente: ");
22            scanf("%f", &nota);
23        }
24        soma=soma+nota;
25        if (nota>maior)
26            maior=nota;
27        if (nota<menor)
28            menor=nota;
29        cont++;
30    }while (cont<=10);
31    // Apresentação de Resultados
32    printf("\nA média das notas fornecidas é: %.2f", (soma/10));
33    printf("\nA soma das notas é: %.2f", soma);
34    printf("\nA maior nota é: %.2f", maior);
35    printf("\nA menor nota é: %.2f", menor);
36    getch();// Função que aguarda o recebimento de um caractere via teclado
37

```

Figura 8.5 – Código do programa em C: *respexe83.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite 10 notas para calcular a média:
Digite a 1.a nota: 1.5
Digite a 2.a nota: 8
Digite a 3.a nota: 7.5
Digite a 4.a nota: 6
Digite a 5.a nota: 10
Digite a 6.a nota: 9.5
Digite a 7.a nota: 8.5
Digite a 8.a nota: 4.5
Digite a 9.a nota: 5
Digite a 10.a nota: 3

A média das notas fornecidas é: 6.35
A soma das notas é: 63.50
A maior nota é: 10.00
A menor nota é: 1.50

```

Figura 8.6 – Resultado da execução do código do programa em C: *respexe83.cpp*

Fonte: Elaboração própria

8.4 Faça um programa que leia dois conjuntos de dois valores, o primeiro representando o número do aluno e o segundo representando a sua altura em centímetros. Encontre o aluno mais alto e o mais baixo. Mostre o número do aluno mais alto, e o número do aluno mais baixo, juntamente com suas alturas.

Resposta: o código do programa é apresentado na Figura 8.7 e o resultado da execução desse código nas Figuras 8.8 e 8.9.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int cont=1, num, maior_num, menor_num;
7     float altura, menor_altura=3.00, maior_altura=0.0;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12    //Início do programa
13    printf("Você deverá digitar 2 conjuntos de 2 valores, o número e a altura do aluno: \n\n");
14    // Laço para pedir todas informações de todos alunos, e já identificar os maiores e menores.
15    do
16    {
17        printf("Número do %d.o aluno: ", cont);
18        scanf("%d", &num);
19        printf("Altura do %d.o aluno: ", cont);
20        scanf("%f", &altura);
21        if (altura>maior_altura)
22        {
23            maior_altura=altura;
24            maior_num=num;
25        }
26        if (altura<menor_altura)
27        {
28            menor_altura=altura;
29            menor_num=num;
30        }
31        cont++;
32    }while (cont<=2);
33    // Apresentação de resultados
34    printf("\n\nO aluno mais baixo e o de número %d com %.2fm.\n", menor_num, menor_altura);
35    printf("O aluno mais alto e o de número %d com %.2fm.", maior_num, maior_altura);
36    getch();// Função que aguarda o recebimento de um caractere via teclado
37 }
```

Figura 8.7 – Código do programa em C: *respexe84.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Voce devera digitar 2 conjuntos de 2 valores, o numero e a altura do aluno:  
Numero do 1.o aluno: 9  
Altura do 1.o aluno: 1.72  
Numero do 2.o aluno: 3  
Altura do 2.o aluno: 1.56  
  
O aluno mais baixo e o de numero 3 com 1.56m.  
O aluno mais alto e o de numero 9 com 1.72m.
```

Figura 8.8 – Resultado da execução do código do programa em C: *respexe84.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Voce devera digitar 2 conjuntos de 2 valores, o numero e a altura do aluno:  
Numero do 1.o aluno: 20  
Altura do 1.o aluno: 1.95  
Numero do 2.o aluno: 21  
Altura do 2.o aluno: 1.74  
  
O aluno mais baixo e o de numero 21 com 1.74m.  
O aluno mais alto e o de numero 20 com 1.95m.
```

Figura 8.9 – Resultado da execução do código do programa em C: *respexe84.cpp*

Fonte: Elaboração própria

8.5 Faça um programa que mostre todos os números pares existentes entre 1 e 50.

Resposta: o código do programa é apresentado na Figura 8.10 e o resultado da execução desse código na Figura 8.11.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int cont=2;// apenas a variável para manipular o laço de repetição
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início do programa
12    printf("Os números pares entre 1 e 50 são: \n");
13    do
14    {
15        printf("%d ", cont);
16        cont=cont+2;
17    }while (cont<50);
18    getch();// Função que aguarda o recebimento de um caractere via teclado
19 }

```

Figura 8.10 – Código do programa em C: *respexe85.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Os números pares entre 1 e 50 são:
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42
44 46 48

```

Figura 8.11 – Resultado da execução do código do programa em C: *respexe85.cpp*

Fonte: Elaboração própria

8.6 Faça um programa que calcule o valor da soma:

$$S = 1/1 + 3/2 + 5/3 + 7/4 + \dots + 99/50.$$

Resposta: o código do programa é apresentado na Figura 8.12 e o resultado da execução desse código na Figura 8.13.

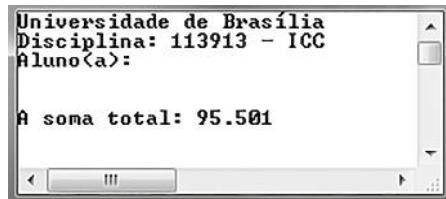
```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h>/Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     float soma=0;
7     float cont1=1, cont2=1;
8     //Cabeçalho
9     printf("Universidade de Brasília\n", 161);
10    printf("Disciplina: 113913 - ICC \n");
11    printf("Aluno(a):\n\n");
12    //Início do programa
13    do
14    {
15        soma=soma+(cont2/cont1); // Calculo da soma
16        cont2=cont2+2; // cont2 cresce 2 vezes mais que i
17        cont1=cont1+1;
18    }while(cont2<=99);
19    /* Tanto cont1 quanto cont2 poderiam ser usadas para o teste no final, contudo para cont2<99 e
20    cont1<50, pois o último fator da soma é 99/50*/
21    printf("A soma total: %.3f", soma);
22    getch();// Função que aguarda o recebimento de um caractere via teclado
23}

```

Figura 8.12 – Código do programa em C: *respexe86.cpp*

Fonte: Elaboração própria

Figura 8.13 – Resultado da execução do código programa em C: *respexe86.cpp*

Fonte: Elaboração própria

- 8.7 Desenvolva um programa que apresente a tabuada de 0 até 9 para o intervalo de 0 até 10. Entre cada tabuada, coloque uma pausa para o usuário conseguir lê-la.

Resposta: o código do programa é apresentado na Figura 8.14 e o resultado da execução desse código na Figura 8.15.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<cconio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int cont=1, auxcont; //Declaração de variáveis
7     //Cabeçalho
8     printf("Universidade de Brasília\n", 161);
9     printf("Disciplina: 113913 - ICC \n");
10    printf("Aluno(a):\n\n");
11    //Início do programa
12    // Laço de repetição para a tabuada
13    do
14    {
15        auxcont=0;
16        do
17        {
18            printf("%d x %d = %d\n", cont, auxcont, cont*auxcont);
19            auxcont++;
20        }while(auxcont<=10);
21        getch();
22        printf("\n");
23        cont++;
24    }while(cont<=9);
25    getch();// Função que aguarda o recebimento de um caractere via teclado
26 }

```

Figura 8.14 – Código do programa em C: *respexe87.cpp*

Fonte: Elaboração própria

1 x 0 = 0	5 x 0 = 0	9 x 0 = 0
1 x 1 = 1	5 x 1 = 5	9 x 1 = 9
1 x 2 = 2	5 x 2 = 10	9 x 2 = 18
1 x 3 = 3	5 x 3 = 15	9 x 3 = 27
1 x 4 = 4	5 x 4 = 20	9 x 4 = 36
1 x 5 = 5	5 x 5 = 25	9 x 5 = 45
1 x 6 = 6	5 x 6 = 30	9 x 6 = 54
1 x 7 = 7	5 x 7 = 35	9 x 7 = 63
1 x 8 = 8	5 x 8 = 40	9 x 8 = 72
1 x 9 = 9	5 x 9 = 45	9 x 9 = 81
1 x 10 = 10	5 x 10 = 50	9 x 10 = 90

Figura 8.15 – Resultado da execução do código do programa em C: *respexe87.cpp*

Fonte: Elaboração própria

8.8 Desenvolva um programa para calcular a tabuada de um intervalo de números que será informado pelo usuário.

Resposta: o código do programa é apresentado na Figura 8.16 e o resultado da execução desse código na Figura 8.17.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 int main (){//Programa principal
5     // Declaração de variáveis:
6     int cont, auxcont, num1, num2;
7     do //Validação alternativa
8     {
9         system("cls");//Função para limpar a tela
10        //Cabeçalho
11        printf("Universidade de Brasília\n", 161);
12        printf("Disciplina: 113913 - ICC \n");
13        printf("Aluno(a):\n\n");
14        //Início do programa
15
16        printf("Digite o primeiro numero: ");
17        scanf("%d", &num1);
18        printf("Digite o segundo numero: ");
19        scanf("%d", &num2);
20    }while(num1>=num2);
21    cont=num1;
22    // Laço de repetição para a tabuada
23    do
24    {
25        auxcont=1;
26        do
27        {
28            printf("\n%d x %d = %d", cont, auxcont, cont*auxcont);
29            auxcont++;
30        }while(auxcont<=10);
31        printf("\n");
32        getch()// Função que aguarda o recebimento de um caractere via teclado
33        cont++;
34    }while(cont<=num2);
35    getch()// Função que aguarda o recebimento de um caractere via teclado

```

Figura 8.16 – Código do programa em C: *respexe88.cpp*

Fonte: Elaboração própria

```

Digite o primeiro numero: 13
Digite o segundo numero: 16
13 x 1 = 13
13 x 2 = 26
13 x 3 = 39
13 x 4 = 52
13 x 5 = 65
13 x 6 = 78
13 x 7 = 91
13 x 8 = 104
13 x 9 = 117
13 x 10 = 130

14 x 1 = 14
14 x 2 = 28
14 x 3 = 42
14 x 4 = 56
14 x 5 = 70

```

Figura 8.17 – Resultado da execução do código do programa em C: *respexe88.cpp*

Fonte: Elaboração própria

Capítulo 9

Funções

Funções

- 9.1 Escreva um programa que solicite dois números do tipo inteiro distintos ao usuário e que apresente na tela o maior deles. Esse programa deve possuir uma função para verificar qual é o maior número.

Resposta: o código do programa é apresentado na Figura 9.1 e o resultado da execução desse código nas Figuras 9.2 e 9.3.

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabecalho
5  void cabecalho(){
6      printf("Universidade de Brasília\n", 161);
7      printf("Disciplina: 113913 - ICC \n");
8      printf("Aluno(a):\n\n");
9  }
10 //Declaração da função verif
11 int verif(int num1, int num2){
12     //Retorna o maior número entre os recebidos
13     if(num1>num2)
14         return num1;
15     else
16         return num2;
17 }
18 int main (){//Programa principal
19     // Declaração de variáveis:
20     int num1,num2;
21     //Chamada do procedimento Cabeçalho
22     cabecalho();
23     //Início do Programa
24     printf("Digite o primeiro numero: ");
25     scanf("%d",&num1);
26     printf("Digite o segundo numero: ");
27     scanf("%d",&num2);
28     //Validação, os números devem ser diferentes:
29     while(num2==num1)
30     {
31         printf("Erro! Os números devem ser diferentes. Digite outro numero: ");
32         scanf("%d",&num2);
33     }
34     //Resultados:
35     printf("O maior numero é: %d\n\n",verif(num1,num2));
36     getch(); // Função que aguarda o recebimento de um caractere via teclado
37 }
```

Figura 9.1 – Código do programa em C: *respexe91.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o primeiro numero: 17
Digite o segundo numero: 25

O maior numero e: 25
```

Figura 9.2 – Resultado da execução do código do programa em C: *respexe91.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o primeiro numero: 165
Digite o segundo numero: 120

O maior numero e: 165
```

Figura 9.3 – Resultado da execução do código do programa em C: *respexe91.cpp*

Fonte: Elaboração própria

9.2 Desenvolva um programa para calcular e comparar a área de dois retângulos A e B, o programa deverá dizer qual retângulo possui a maior área ou se eles possuem tamanhos iguais. Esse programa deve possuir uma função para calcular a área do retângulo. Dados de entrada: tamanho da base e da altura (tipo das variáveis: inteiro, valor em centímetros).

Resposta: o código do programa é apresentado na Figura 9.4 e o resultado da execução desse código na Figura 9.5.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 //Declaração da função verif e da função calc_area
11 int verif(int num1, int num2){
12     //Retorna o maior número entre os recebidos
13     if(num1>num2)
14         return num1;
15     else
16         return num2;
17 }
18 int calc_area(int lado,int altura){
19     return lado*altura;// Função que retorna o cálculo da área
20 }
21 int main (){//Programa principal
22     // Declaração de variáveis:
23     int lado,altura,area1,area2;
24     //Chamada do procedimento Cabeçalho
25     cabecalho();
26     //Início do Programa
27     printf("Digite a altura do primeiro retângulo: ");
28     scanf("%d", &altura);
29     printf("Digite o lado do primeiro retângulo: ");
30     scanf("%d", &lado);
31     area1=calc_area(lado,altura); //Cálculo da área através da função
32     printf("Digite a altura do segundo retângulo: ");
33     scanf("%d", &altura);
34     printf("Digite o lado do segundo retângulo: ");
35     scanf("%d", &lado);
36     area2=calc_area(lado,altura); //Cálculo da área através da função
37     //Resultados
38     printf("\nÁrea do primeiro retângulo: %d\n",area1);
39     printf("Área do segundo retângulo: %d\n",area2);
40     if(area1==area2)
41         printf("As áreas são iguais.");
42     else
43         printf("A maior área é: %d,verif(area1, area2)); //Usando a função do exercício anterior
44     getch();// Função que aguarda o recebimento de um caractere via tecla
45 }

```

Figura 9.4 – Código do programa em C: *respexe92.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite a altura do primeiro retangulo: 13
Digite o lado do primeiro retangulo: 15
Digite a altura do segundo retangulo: 10
Digite o lado do segundo retangulo: 20  
  
Area do primeiro retangulo: 195
Area do segundo retangulo: 200
A maior area e: 200
```

Figura 9.5 – Resultado da execução do código do programa em C: *respexe92.cpp*

Fonte: Elaboração própria

- 9.3 Escreva um programa que solicite a temperatura em Celsius ao usuário e apresente na tela o resultado da conversão dessa temperatura em Fahrenheit. Esse programa deve possuir uma função para converter a temperatura. Dados: $Fahrenheit = Celsius \times 1,8 + 32$.

Resposta: o código do programa é apresentado na Figura 9.6 e o resultado do código da execução desse código nas Figuras 9.7, 9.8 e 9.9.

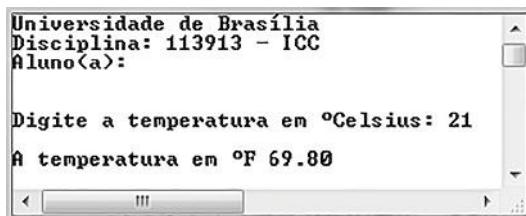
```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabeçalho
5  void cabecalho(){
6      printf("Universidade de Brasília\n", 161);
7      printf("Disciplina: 113913 - ICC \n");
8      printf("Aluno(a):\n\n");
9  }
10 //Declaração da função converter
11 float converter(float tempcel){
12     //Definição do corpo da função
13     float tempfar;
14     tempfar=(tempcel*1.8)+32;
15     return tempfar;
16 }
17 int main (){//Programa principal
18     // Declaração de variáveis:
19     float tempfar,tempcel;
20     //Chamada do procedimento Cabeçalho
21     cabecalho();
22     //Início do Programa
23     printf("Digite a temperatura em °Celsius: ", 248);
24     scanf("%f",&tempcel);
25     tempfar=converter(tempcel);//Converte a temperatura de C para F
26     //Resultados:
27     printf("\nA temperatura em °F %.2f", 248, tempfar);
28     getch(); // Função que aguarda o recebimento de um caractere via tecla
29 }

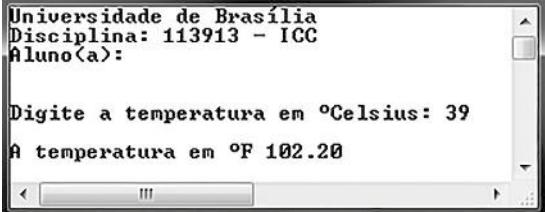
```

Figura 9.6 – Código do programa em C: *respexe93.cpp*

Fonte: Elaboração própria

Figura 9.7 – Resultado da execução do código do programa em C: *respexe93.cpp*

Fonte: Elaboração própria

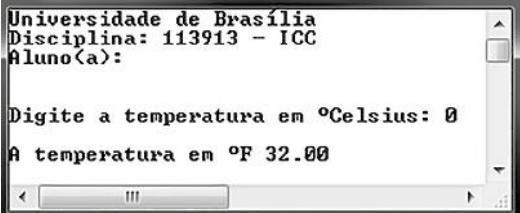


```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite a temperatura em °Celsius: 39
A temperatura em °F 102.20
```

Figura 9.8 – Resultado da execução do código do programa em C: respexe93.cpp

Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite a temperatura em °Celsius: 0
A temperatura em °F 32.00
```

Figura 9.9 – Resultado da execução do código do programa em C: respexe93.cpp

Fonte: Elaboração própria

- 9.4 Escreva um programa que solicite dois números ao usuário e apresente na tela o resultado da soma do módulo desses números. Esse programa deve possuir uma função para calcular o módulo.

Resposta: o código do programa é apresentado na Figura 9.10 e o resultado da execução desse código nas Figuras 9.11, 9.12, 9.13 e 9.14.

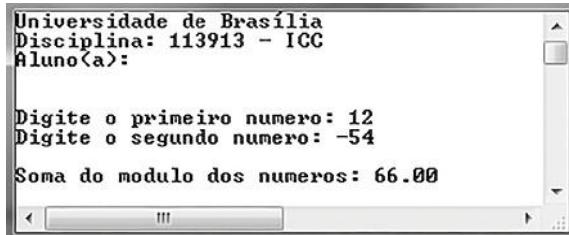
```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 float soma(float num1,float num2){
11     //Função retorna a soma dos números recebidos
12     return num1+num2;
13 }
14 float mod(float num){
15     //Função que retorna o módulo
16     if(num>0)
17         return num;
18     else
19         return -num;
20 }
21 int main (){//Programa principal
22     // Declaração de variáveis:
23     float num1,num2;
24     //Chamada do procedimento Cabeçalho
25     cabecalho();
26     //Início do Programa
27     printf("Digite o primeiro número: ");
28     scanf("%f",&num1);
29     printf("Digite o segundo número: ");
30     scanf("%f",&num2);
31     //Os números são convertidos em seus módulos
32     num1=mod(num1);
33     num2=mod(num2);
34     //Resultado por meio da função soma da ParteB
35     printf("Soma do módulo dos números: %.2f\n\n",soma(num1,num2));
36     getch(); // Função que aguarda o recebimento de um caractere via teclado
37 }

```

Figura 9.10 – Código do programa em C: *respexe94.cpp*

Fonte: Elaboração própria

Figura 9.11 – Resultado da execução do código do programa em C: *respexe94.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o primeiro numero: -10
Digite o segundo numero: -14
Soma do modulo dos numeros: 24.00
```

Figura 9.12 – Resultado da execução do código do programa em C: *respexe94.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o primeiro numero: -7
Digite o segundo numero: 8
Soma do modulo dos numeros: 15.00
```

Figura 9.13 – Resultado da execução do código do programa em C: *respexe94.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o primeiro numero: 18
Digite o segundo numero: 20
Soma do modulo dos numeros: 38.00
```

Figura 9.14 – Resultado da execução do código do programa em C: *respexe94.cpp*

Fonte: Elaboração própria

- 9.5 Desenvolva um programa que solicite três notas de provas de um aluno e apresente na tela a média dessas notas. A obtenção das notas deve ser realizada na função principal e o cálculo da média das notas deve

ser obtido por outra função (MEDIA). Para o cálculo da média considere que a primeira prova tem peso 1 e as outras duas provas possuem peso 2.

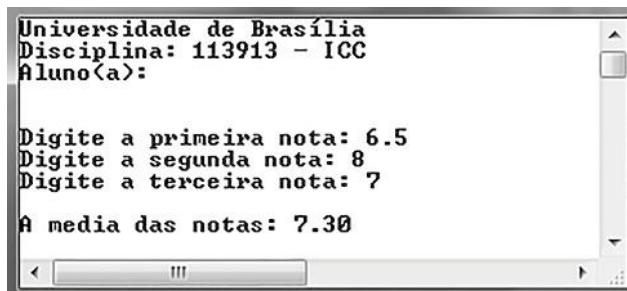
Resposta: o código do programa é apresentado na Figura 9.15 e o resultado da execução desse código nas Figuras 9.16 e 9.17.

```

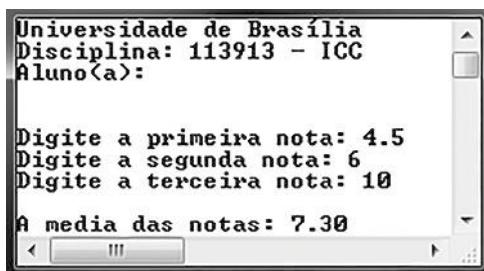
1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n\n");
9 }
10 float media(float nota1, float nota2, float nota3){
11     //Calculo da media com b e c com peso 2
12     nota2=2*nota2;
13     nota3=2*nota3;
14     return ((nota1+nota2+nota3)/5);
15 }
16 int main (){//Programa principal
17     // Declaração de variáveis:
18     float nota1,nota2,nota3;
19     //Chamada do procedimento Cabeçalho
20     cabecalho();
21     //Início do Programa
22     do
23     {
24         printf("Digite a primeira nota: ");
25         scanf("%f", &nota1);
26     }while(nota1<0 || nota1>10);
27     do
28     {
29         printf("Digite a segunda nota: ");
30         scanf("%f", &nota2);
31     }while(nota2<0 || nota2>10);
32     do
33     {
34         printf("Digite a terceira nota: ");
35         scanf("%f", &nota3);
36     }while(nota3<0 || nota3>10);
37     //Calculo da media das notas
38     printf("\nA media das notas: %.2f\n\n",media(nota1,nota2,nota3));
39     getch(); // Função que aguarda o recebimento de um caractere via teclado
40 }
```

Figura 9.15 – Código do programa em C: *respexe95.cpp*

Fonte: Elaboração própria

Figura 9.16 – Resultado da execução do código do programa em C: *respexe95.cpp*

Fonte: Elaboração própria

Figura 9.17 – Resultado da execução do código do programa em C: *respexe95.cpp*

Fonte: Elaboração própria

9.6 Escreva um programa que solicite dois números inteiros ao usuário e que apresente na tela como resultado o dobro desses números que devem ser somados e o resultado da soma devem ser triplicados. Esse programa deve possuir uma função para dobrar o valor de um número, outra para somar dois números e uma terceira para triplicar um número.

Resposta: o código do programa é apresentado na Figura 9.18 e o resultado da execução desse código nas Figuras 9.19 e 9.20.

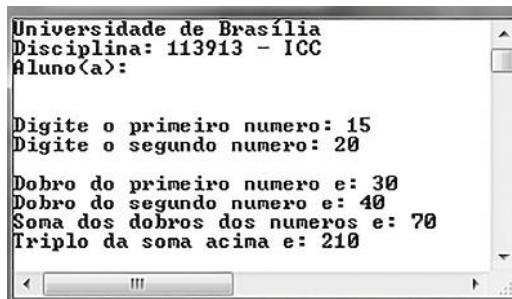
```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h>//Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 //Declaração da função das funções soma, dobro e triplica
11 int soma(int num1,int num2){
12     return num1+num2;//Função retorna a soma dos números recebidos
13 }
14 int dobro(int num){
15     return 2*num; //Função retorna o dobro do número recebido
16 }
17 int triplica(int num){
18     return 3*num; //Função retorna o triplo do número recebido
19 }
20 int main (){//Programa principal
21     // Declaração de variáveis:
22     int num1,num2,sum;
23     //Chamada do procedimento Cabeçalho
24     cabecalho();
25     //Início do Programa
26     printf("Digite o primeiro numero: ");
27     scanf("%d",&num1);
28     printf("Digite o segundo numero: ");
29     scanf("%d",&num2);
30     //Resultados por meio das funções
31     printf("\nDobro do primeiro numero e: %d\nDobro do segundo numero e: %d\n",dobra(num1),dobra(num2));
32     sum=soma(dobro(num1),dobra(num2));
33     printf("Soma dosdobros dos numeros e: %d\n",sum);
34     printf("Triplo da soma acima e: %d\n",triplica(sum));
35     getch();// Função que aguarda o recebimento de um caractere via teclado
36 }

```

Figura 9.18 – Código do programa em C: *respexe96.cpp*

Fonte: Elaboração própria

Figura 9.19 – Resultado da execução do código do programa em C: *respexe96.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília  
Disciplina: 113913 - ICC  
Aluno(a):  
  
Digite o primeiro numero: 7  
Digite o segundo numero: 9  
  
Dobro do primeiro numero e: 14  
Dobro do segundo numero e: 18  
Soma dosdobros dos numeros e: 32  
Triplo da soma acima e: 96
```

Figura 9.20 – Resultado da execução do programa em C: *respexe96.cpp*

Fonte: Elaboração própria

- 9.7 Desenvolva um programa que solicite ao usuário idade de três pessoas e apresente na tela a maior idade. Esse programa deve possuir uma função para verificar qual é a maior idade.

Resposta: o código do programa é apresentado na Figura 9.21 e o resultado da execução desse código nas Figuras 9.22 e 9.23.

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabeçalho
5  void cabecalho(){
6      printf("Universidade de Brasília\n", 161);
7      printf("Disciplina: 113913 - ICC \n");
8      printf("Aluno(a):\n\n");
9  }
10 int maior(int idd1,int idd2,int idd3){
11     //Condições para retornar o maior valor
12     if(idd1==idd2 && idd2==idd3)
13         return 0;
14     else if(idd1>=idd2 && idd1>=idd3)
15         return idd1;
16     else if(idd2>=idd3)
17         return idd2;
18     else
19         return idd3;
20 }
21 int main (){//Programa principal
22     // Declaração de variáveis:
23     int idd1,idd2,idd3,maior_idd;
24     //Chamada do procedimento Cabeçalho
25     cabecalho();
26     //Início do Programa
27     printf("Digite a primeira idade: ");
28     scanf("%d",&idd1);
29     printf("Digite a segunda idade: ");
30     scanf("%d",&idd2);
31     printf("Digite a terceira idade: ");
32     scanf("%d",&idd3);
33     //Maior idade
34     maior_idd=maior(idd1,idd2,idd3);
35     if(maior_idd==0)
36         printf("\nAs idades são iguais: %d\n\n",idd1);
37     else
38         printf("\nMaior idade: %d\n\n",maior_idd);
39     getch(); // Função que aguarda o recebimento de um caractere via teclado
40 }

```

Figura 9.21 – Código do programa em C: *respexe97.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite a primeira idade: 19
Digite a segunda idade: 43
Digite a terceira idade: 22  
  
Maior idade: 43
```

Figura 9.22 – Resultado da execução do código do programa em C: *respexe97.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite a primeira idade: 63
Digite a segunda idade: 32
Digite a terceira idade: 9  
  
Maior idade: 63
```

Figura 9.23 – Resultado da execução do código do programa em C: *respexe97.cpp*

Fonte: Elaboração própria

9.8 Desenvolva um programa que solicite ao usuário a idade de três pessoas e apresente na tela a maior e a menor idade. Esse programa deve possuir uma função para verificar qual é a maior idade e outra para verificar a menor idade.

Resposta: o código do programa é apresentado na Figura 9.24 e o resultado da execução desse código nas Figuras 9.25 e 9.26.

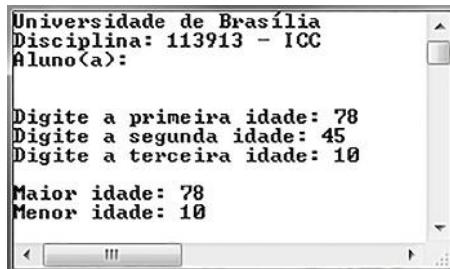
```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabecalho
5  void cabecalho(){
6      printf("Universidade de Brasília\n", 161);
7      printf("Disciplina: 113913 - ICC \n");
8      printf("Aluno(a):\n\n");
9  }
10 //Declaração da função maior e menor
11 int maior(int idd1,int idd2,int idd3){
12     //Condições para retornar o maior valor
13     if(idd1==idd2 && idd2==idd3)
14         return 0;
15     else if(idd1>=idd2 && idd1>=idd3)
16         return idd1;
17     else if(idd2>=idd3)
18         return idd2;
19     else
20         return idd3;
21 }
22 int menor(int idd1,int idd2,int idd3){
23     //Condições para retornar o menor valor
24     if(idd1==idd2 && idd2==idd3)
25         return 0;
26     else if(idd1<=idd2 && idd1<=idd3)
27         return idd1;
28     else if(idd2<=idd3)
29         return idd2;
30     else
31         return idd3;
32 }
33
34 int main ()//Programa principal
35     // Declaração de variáveis:
36     int idd1,idd2,idd3,maior_idd,menor_idd;
37     //Chamada do procedimento Cabeçalho
38     cabecalho();
39     //Início do Programa
40     printf("Digite a primeira idade: ");
41     scanf("%d",&idd1);
42     printf("Digite a segunda idade: ");
43     scanf("%d",&idd2);
44     printf("Digite a terceira idade: ");
45     scanf("%d",&idd3);
46     //Maior e menor idade
47     maior_idd=maior(idd1,idd2,idd3);
48     menor_idd=menor(idd1,idd2,idd3);
49     if(maior_idd==0 && menor_idd==0)
50         printf("\nAs idades são iguais: %d\n\n",idd1);
51     else
52     {
53         printf("\nMaior idade é: %d\n",maior_idd);
54         printf("Menor idade é: %d\n\n",menor_idd);
55     }
56     getch(); // Função que aguarda o recebimento de um caractere via teclado
57 }

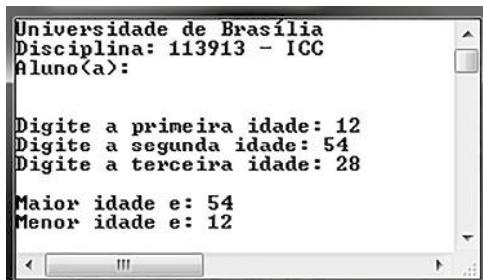
```

Figura 9.24 – Código do programa em C: *respexe98.cpp*

Fonte: Elaboração própria

Figura 9.25 – Resultado da execução do código do programa em C: *respexe98.cpp*

Fonte: Elaboração própria

Figura 9.26 – Resultado da execução do código do programa em C: *respexe98.cpp*

Fonte: Elaboração própria

- 9.9 Escreva um programa que leia os dados de um funcionário e que aplique um aumento sobre o seu salário, sendo que a empresa definiu um aumento de 10% para quem possuir mais de cinco anos de casa e for casado, para os demais o aumento é de 8%. Esse aumento deve ser calculado por uma função. Dados de entrada: salário bruto, quantidades de anos na empresa, e estado civil (C, c, S, s). Dados de saída: taxa do aumento, salário inicial, e salário com aumento do funcionário.

Resposta: o código do programa é apresentado na Figura 9.27 e o resultado da execução desse código nas Figuras 9.28 e 9.29.

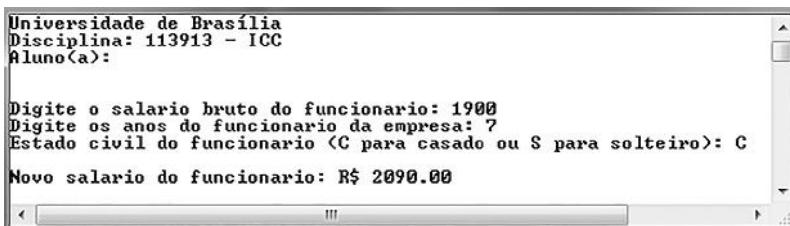
```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabecalho
5  void cabecalho(){
6      printf("Universidade de Brasília\n", 161);
7      printf("Disciplina: 113913 - ICC \n");
8      printf("Aluno(a):\n\n");
9  }
10 //Declaração da função reajuste_sal
11 float reajuste_sal(int anos, float sal, char est_civ){
12     //Cálculo de acordo com as condições apresentadas no problema
13     if((anos >=5) && (est_civ=='C' || est_civ=='c'))
14         return sal*1.10;
15     else
16         return sal*1.08;
17 }
18 int main (){//Programa principal
19     // Declaração de variáveis:
20     int anos;
21     float sal_brto,novo_sal;
22     char est_civil;|
23     //Chamada do procedimento Cabeçalho
24     cabecalho();
25     //Início do Programa
26     do
27     {
28         printf("Digite o salário bruto do funcionário: ");
29         scanf("%f",&sal_brto);
30     }while(sal_brto <= 0); //validação
31     do
32     {
33         printf("Digite os anos de serviço da empresa: ");
34         scanf("%d",&anos);
35     }while(anos <= 0); //validação
36     printf("Estado civil do funcionário (C para casado ou S para solteiro): ");
37     fflush(stdin); //Função para limpar a memória do teclado
38     scanf("%c",&est_civil);
39     while(est_civil!='c' && est_civil!='C' && est_civil!='S' && est_civil!='s')
40     {
41         printf("Digite um estado civil válido (C/S): ");
42         fflush(stdin);
43         scanf("%c",&est_civil);
44     } // validação de est_civil
45     //Resultados
46     novo_sal=reajuste_sal(anos, sal_brto, est_civil);
47     printf("\nNovo salário do funcionário: R$ %.2f\n\n",novo_sal);
48     getch(); // Função que aguarda o recebimento de um caractere via teclado
49 }

```

Figura 9.27 – Código do programa em C: *respexe99.cpp*

Fonte: Elaboração própria

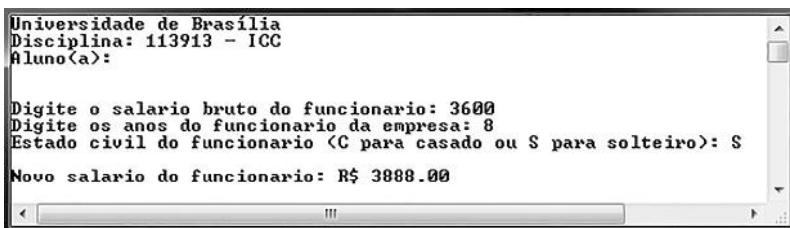


```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite o salario bruto do funcionario: 1900
Digite os anos do funcionario da empresa: 7
Estado civil do funcionario (C para casado ou S para solteiro): C
Novo salario do funcionario: R$ 2090.00
```

Figura 9.28 – Resultado da execução do código do programa em C: *respexe99.cpp*

Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite o salario bruto do funcionario: 3600
Digite os anos do funcionario da empresa: 8
Estado civil do funcionario (C para casado ou S para solteiro): S
Novo salario do funcionario: R$ 3888.00
```

Figura 9.29 – Resultado da execução do código do programa em C: *respexe99.cpp*

Fonte: Elaboração própria

- 9.10 Escreva um programa que solicite dois números ao usuário e que apresente na tela o resultado da sua soma e o dobro de cada um deles. Esse programa deve possuir duas funções, uma para calcular a soma e outro para calcular o dobro desses números.

Resposta: o código do programa é apresentado na Figura 9.30 e o resultado da execução desse código nas Figuras 9.31 e 9.32.

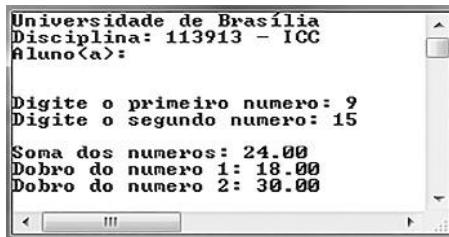
```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h>//Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabecalho|
5  void cabecalho(){
6      printf("Universidade de Brasília\n", 161);
7      printf("Disciplina: 113913 - ICC \n");
8      printf("Aluno(a):\n\n");
9  }
10
11 float soma(float num1,float num2){
12     return num1+num2;//Função retorna a soma dos números recebidos
13 }
14
15 float dobro(float num){
16     return 2*num;//Função retorna o dobro do número recebido
17 }
18
19 int main (){//Programa principal
20     // Declaração de variáveis:
21     float num1,num2,sum, dbl1,dbl2;
22     //Chamada do procedimento Cabeçalho
23     cabecalho();
24     //Início do Programa
25     printf("Digite o primeiro numero: ");
26     scanf("%f",&num1);
27     printf("Digite o segundo numero: ");
28     scanf("%f",&num2);
29     sum=soma(num1,num2);//Soma dos dois números
30     dbl1=dobro(num1);//Dobro do primeiro número
31     dbl2=dobro(num2);//Dobro do segundo número
32     //Apresentação dos resultados
33     printf("Soma dos números: %.2f\n",sum);
34     printf("Dobro do numero 1: %.2f\nDobro do numero 2: %.2f\n\n",dbl1,dbl2);
35     getch();// Função que aguarda o recebimento de um caractere via teclado
36 }

```

Figura 9.30 – Código do programa em C: *respexe910.cpp*

Fonte: Elaboração própria

Figura 9.31 – Resultado da execução do código do programa em C: *respexe910.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o primeiro numero: 100
Digite o segundo numero: 200

Soma dos numeros: 300.00
Dobro do numero 1: 200.00
Dobro do numero 2: 400.00
```

Figura 9.32 – Resultado da execução do código do programa em C: *respexe910.cpp*

Fonte: Elaboração própria

- 9.11 Desenvolva um programa, *tabuada.cpp*, que apresente a tabuada de 0 até 9 para um número determinado pelo usuário. Esse programa deve conter um procedimento que recebe por parâmetro o número digitado pelo usuário. Esse procedimento deve escrever as tabuadas na tela.

Resposta: o código do programa é apresentado na Figura 9.33 e o resultado da execução desse código na Figura 9.34.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 void imprimetabu(int num){
11     int cont;
12     for(cont=0;cont<=9;cont++)
13         printf("\n%d x %d = %d", num, cont, num*cont);
14 }
15 int main (){//Programa principal
16     // Declaração de variáveis:
17     int num;
18     //Chamada do procedimento Cabeçalho
19     cabecalho();
20     //Início do Programa
21     printf("Digite de qual numero deseja a tabuada: ");
22     scanf("%d", &num);
23     // Invocação do procedimento
24     imprimetabu(num);
25     getch(); // Função que aguarda o recebimento de um caractere via teclado
26 }

```

Figura 9.33 – Código do programa em C: *respxe911.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite de qual numero deseja a tabuada: 23
23 x 0 = 0
23 x 1 = 23
23 x 2 = 46
23 x 3 = 69
23 x 4 = 92
23 x 5 = 115
23 x 6 = 138
23 x 7 = 161
23 x 8 = 184
23 x 9 = 207

```

Figura 9.34 – Resultado da execução do código do programa em C: *respxe911.cpp*

Fonte: Elaboração própria

- 9.12 Desenvolva um programa para calcular a tabuada de um intervalo de dois números que será informado pelo usuário. Entre cada tabuada, deve haver uma pausa para o usuário conseguir lê-la. Esse programa deve conter um procedimento, imptabuinter, que recebe por parâmetro os dois números do intervalo. Esse procedimento deve escrever as tabuadas na tela.

Resposta: o código do programa é apresentado na Figura 9.35 e o resultado da execução desse código na Figura 9.36.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabeçalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n\n");
9 }
10 void imprimetabu(int num1, int num2){
11     int cont1,cont2;
12     for(cont1=num1;cont1<=num2;cont1++)
13     {
14         for(cont2=0; cont2<=9; cont2++)
15             printf("\n%d x %d = %d", cont1, cont2, cont1*cont2);
16         printf("\n");
17         getch(); //Função para aguardar o recebimento de qualquer tecla
18     }
19 }
20 int main (){//Programa principal
21     // Declaração de variáveis:
22     int num1, num2;
23     //Chamada do procedimento Cabeçalho
24     cabecalho();
25     //Início do Programa
26     printf("Digite o primeiro numero: ");
27     scanf("%d", &num1);
28     printf("Digite o segundo numero: ");
29     scanf("%d", &num2);
30     if(num1<num2) // Invocação do procedimento para num1<num2
31         imprimetabu(num1, num2);
32     else if(num2<num1) // Invocação do procedimento para num2<num1
33         imprimetabu(num2, num1);
34     getch(); // Função que aguarda o recebimento de um caractere via teclado
35 }
```

Figura 9.35 – Código do programa em C: *respexe912.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o primeiro numero: 20
Digite o segundo numero: 22

20 x 0 = 0
20 x 1 = 20
20 x 2 = 40
20 x 3 = 60
20 x 4 = 80
20 x 5 = 100
20 x 6 = 120
20 x 7 = 140
20 x 8 = 160
20 x 9 = 180

21 x 0 = 0
21 x 1 = 21
21 x 2 = 42
21 x 3 = 63
21 x 4 = 84
21 x 5 = 105
21 x 6 = 126
21 x 7 = 147
21 x 8 = 168
21 x 9 = 189

22 x 0 = 0
22 x 1 = 22
22 x 2 = 44
22 x 3 = 66
22 x 4 = 88
22 x 5 = 110
22 x 6 = 132
22 x 7 = 154
22 x 8 = 176
22 x 9 = 198

```



```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o primeiro numero: 2
Digite o segundo numero: 4

2 x 0 = 0
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18

3 x 0 = 0
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27

4 x 0 = 0
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36

```

Figura 9.36 – Resultado da execução do código do programa em C: *respxexe912.cpp*

Fonte: Elaboração própria

- 9.13 Desenvolva um programa para calcular e imprimir o volume de uma esfera. Esse programa deva possuir uma função que recebe por parâmetro o raio de uma esfera e calcula o seu volume ($v = (4/3) \cdot \pi \cdot R^3$).

Resposta: o código do programa é apresentado na Figura 9.37 e o resultado da execução desse código nas Figuras 9.38 e 9.39.

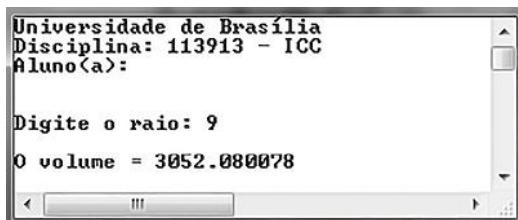
```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabeçalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 float calc(int raio){
11     float vol;
12     vol=(4*3.14*raio*raio*raio)/3;
13     return vol;
14 }
15 int main (){//Programa principal
16     // Declaração de variáveis:
17     int raio;
18     //Chamada do procedimento Cabeçalho
19     cabecalho();
20     //Início do Programa
21     do
22     {
23         printf("Digite o raio: ");
24         scanf("%d", &raio);
25         }while(raio<0);
26         //resultados
27         printf("\nO volume = %f\n", calc(raio));
28         getch(); // Função que aguarda o recebimento de um caractere via teclado
29     }

```

Figura 9.37 – Código do programa em C: *respexe913.cpp*

Fonte: Elaboração própria

Figura 9.38 – Resultado da execução do código do programa em C: *respexe913.cpp*

Fonte: Elaboração própria

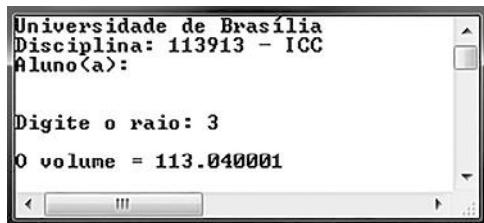


Figura 9.39 – Resultado da execução do código do programa em C: *respexe913.cpp*

Fonte: Elaboração própria

- 9.14 Desenvolva um programa que possua uma função que recebe, por parâmetro, a altura (alt) e o sexo de uma pessoa e retorna o peso ideal. Para homens, calcular usando a fórmula peso ideal = $72.7 \times \text{alt} - 58$ e, para mulheres, peso ideal = $62.1 \times \text{alt} - 44.7$. Depois a função principal deve informar ao usuário o resultado.

Resposta: o código do programa é apresentado na Figura 9.40 e o resultado da execução desse código nas Figuras 9.41 e 9.42.

```
1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 float calc(float alt, char sexo){
11     float peso;
12     if(sexo=='m')
13         peso=(72.7*alt)-58;
14     if(sexo=='f')
15         peso=(62.1*alt)-44.7;
16     return peso;
17 }
18 int main (){//Programa principal
19     // Declaração de variáveis:
20     float alt;
21     char sexo;
22     //Chamada do procedimento Cabecalho
23     cabecalho();
24     //Início do Programa
25     do
26     {
27         printf("Digite m para masc. e f para femin.: ");
28         scanf("%c", &sexo);
29     }while((sexo!='m')&&(sexo!='f'));
30
31     do
32     {
33         printf("Digite a altura: ");
34         scanf("%f", &alt);
35     }while(alt<0);
36     // apresentação de resultados
37     printf("\nO peso ideal = %.2f\n\n", calc(alt,sexo));
38     getch(); // Função que aguarda o recebimento de um caractere via teclado
39 }
```

Figura 9.40 – Código do programa em C: *respexe914.cpp*

Fonte: Elaboração própria

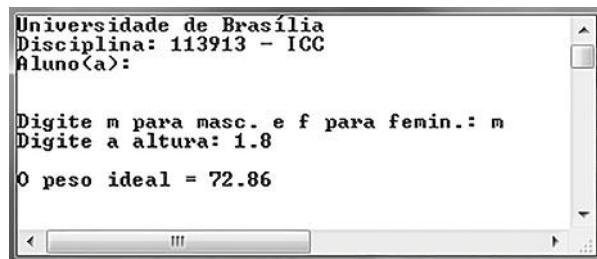


Figura 9.41 – Resultado da execução do código do programa em C: *respexe914.cpp*

Fonte: Elaboração própria

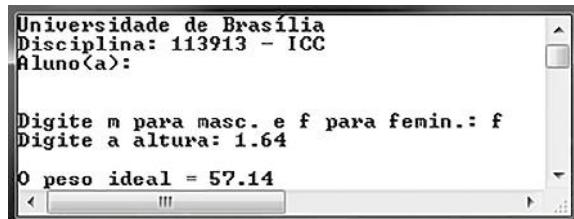


Figura 9.42 – Resultado da execução do código do programa em C: *respexe914cpp*

Fonte: Elaboração própria

- 9.15 Desenvolva um programa que possua um procedimento que recebe três notas de um aluno por parâmetro e uma letra. Se a letra for A, o procedimento calcula a média aritmética das notas do aluno, se for P, calcula a sua média ponderada (pesos: 5, 3 e 2), e se for H, a sua média harmônica.

Resposta: o código do programa é apresentado na Figura 9.43 e o resultado da execução desse código nas Figuras 9.44, 9.45 e 9.46.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabeçalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 float calc_media(float nota1, float nota2, float nota3, char opcao){
11     //Função que retorna a média solicitada
12     float media=0;
13     if(opcao=='A')
14         media=(nota1+nota2+nota3)/3;
15     else if(opcao=='P')
16         media= ((nota1*5)+(nota2*3)+(nota3*2))/10;
17     else if(opcao=='H')
18         media=((1/nota1)+(1/nota2)+(1/nota3))/3;
19     return media;
20 }
21 int main (){//Programa principal
22     // Declaração de variáveis:
23     float nota1, nota2, nota3;
24     char escolha;
25     //Chamada do procedimento Cabeçalho
26     cabecalho();
27     //Início do Programa
28     do
29     {
30         printf("Digite a primeira nota: ");
31         scanf("%f", &nota1);
32     }while(nota1<0 || nota1>10);
33     do
34     {
35         printf("Digite a segunda nota: ");
36         scanf("%f", &nota2);
37     }while(nota2<0 || nota2>10);
38     do
39     {
40         printf("Digite a terceira nota: ");
41         scanf("%f", &nota3);
42     }while(nota3<0 || nota3>10);

```

Figura 9.43 – Código do programa em C: *respexe915.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília  
Disciplina: 113913 - ICC  
Aluno(a):  
  
Digite a primeira nota: 9.5  
Digite a segunda nota: 6.5  
Digite a terceira nota: 8  
Digite a opção para a média:  
Aritimética....A  
Ponderada.....P  
Harmonica.....H  
Opção: P  
  
A média solicitada é igual a 8.30
```

Figura 9.44 – Resultado da execução do código do programa em C: *respexe915.cpp*

Fonte: Elaboração própria

```
Digite a primeira nota: 9.5  
Digite a segunda nota: 6.5  
Digite a terceira nota: 8  
Digite a opção para a média:  
Aritimética....A  
Ponderada.....P  
Harmonica.....H  
Opção: A  
  
A média solicitada é igual a 8.00
```

Figura 9.45 – Resultado da execução do código do programa em C: *respexe915.cpp*

Fonte: Elaboração própria

```
Digite a primeira nota: 9.5  
Digite a segunda nota: 6.5  
Digite a terceira nota: 8  
Digite a opção para a média:  
Aritimética....A  
Ponderada.....P  
Harmonica.....H  
Opção: H  
  
A média solicitada é igual a 0.13
```

Figura 9.46 – Resultado da execução do código do programa em C: *respexe915.cpp*

Fonte: Elaboração própria

9.16 Desenvolva um programa que possua uma função que recebe um valor inteiro e verifica se o valor é positivo ou negativo. A função deve retornar um valor booleano. Depois a função principal deve informar ao usuário o resultado.

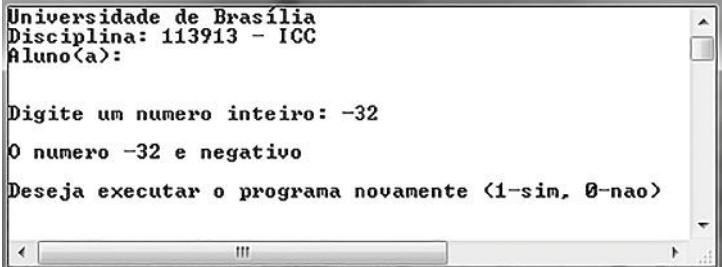
Resposta: o código do programa é apresentado na Figura 9.47 e o resultado da execução desse código nas Figuras 9.48 e 9.49.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabeçalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 int booleana(int num){
11     //Função que retorna valor booleano
12     if(num>0)
13         return 1;
14     else
15         return 0;
16 }
17 int main (){//Programa principal
18     // Declaração de variáveis:
19     int num, opcao;
20     do
21     {
22         system("cls");//Função para limpar a tela
23         //Chamada do procedimento Cabeçalho
24         cabecalho();
25         //Início do Programa
26         printf("Digite um número inteiro: ");
27         scanf("%d", &num);
28         if(booleana(num)==1)
29             printf("\nO número %d é positivo", num);
30         else if(booleana(num)==0)
31             printf("\nO número %d é negativo", num);
32         do
33         {
34             fflush(stdin);
35             printf("\n\nDeseja executar o programa novamente (1-sim, 0-nao): ");
36             scanf("%d", &opcao);
37             }while(opcao==1); opcao!=0;
38     }while(opcao==0);
39 }
```

Figura 9.47 – Código do programa em C: *respexe916.cpp*

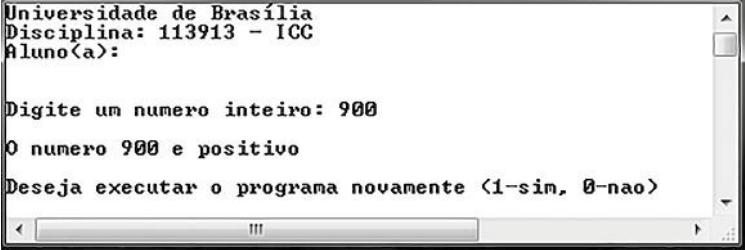
Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite um numero inteiro: -32
O numero -32 é negativo
Deseja executar o programa novamente <1-sim, 0-nao>
```

Figura 9.48 – Resultado da execução do código do programa em C: *respexe916.cpp*

Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite um numero inteiro: 900
O numero 900 é positivo
Deseja executar o programa novamente <1-sim, 0-nao>
```

Figura 9.49 – Resultado da execução do código do programa em C: *respexe916.cpp*

Fonte: Elaboração própria

Capítulo 10

Vetores

Vetores: exercícios

- 10.1 Escreva um programa para informar o maior elemento de um vetor de 5 posições do tipo inteiro. Esse programa deve possuir uma função para verificar o maior número desse vetor.

Resposta: o código do programa é apresentado na Figura 10.1 e o resultado da execução desse código nas Figuras 10.2 e 10.3.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabeçalho
5 void cabecalho()
6 {
7     printf("Universidade de Brasília\n", 161);
8     printf("Disciplina: 113913 - ICC \n");
9     printf("\nAluno(a):\n\n\n");
10 }
11 //função para verificação do maior número do vetor
12 float verMaior(float maior_num[5]) {
13     float maior=0;
14     int cont;
15     for(cont=0;cont<5;cont++)
16     {
17         //Teste verificação do maior número
18         if(maior_num[cont]>maior)
19             maior=maior_num[cont];
20     }
21     return maior;
22 }
23 int main ()//Programa principal
24 {
25     // Declaração de variáveis:
26     float num[5], maior_num;
27     int cont;
28     //Chamada do procedimento Cabeçalho
29     cabecalho();
30     //Início do Programa
31     //Armazenando elementos do vetor
32     for(cont=0;cont<5;cont++)
33     {
34         printf("Digite o %d.o elemento para o vetor: ",cont+1);
35         scanf("%f",&num[cont]);
36     }
37     //Recebendo o retorno da função
38     maior_num=verMaior(num);
39     //Resultados
40     printf("\nO maior numero é: %.2f", maior_num);
41     getch(); // Função que aguarda o recebimento de um caractere via teclado
42 }
```

Figura 10.1 – Código do programa em C: *respexe101.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite o 1.o elemento para o vetor: 7
Digite o 2.o elemento para o vetor: 89
Digite o 3.o elemento para o vetor: -15
Digite o 4.o elemento para o vetor: 4
Digite o 5.o elemento para o vetor: 0  
  
O maior numero e: 89.00
```

Figura 10.2 – Resultado da execução do código do programa em C: respexe101.cpp

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite o 1.o elemento para o vetor: 1
Digite o 2.o elemento para o vetor: 2
Digite o 3.o elemento para o vetor: 3
Digite o 4.o elemento para o vetor: 4
Digite o 5.o elemento para o vetor: 5  
  
O maior numero e: 5.00
```

Figura 10.3 – Resultado da execução do código do programa em C: respexe101.cpp

Fonte: Elaboração própria

- 10.2 Desenvolva um programa que calcule a média das notas de alunos para uma turma de no máximo 100 alunos. O programa deve solicitar no início o tamanho da turma.

Resposta: o código do programa é apresentado na Figura 10.4 e o resultado da execução desse código nas Figuras 10.5 e 10.6.

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h>/Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabecalho
5  void cabecalho(){
6      printf("Universidade de Brasília\n", 161);
7      printf("Disciplina: 113913 - ICC \n");
8      printf("Aluno(a):\n\n");
9  }
10 int main (){//Programa principal
11     // Declaração de variáveis:
12     int qnt, cont;
13     float notas[100], soma=0, media;
14     //Chamada do procedimento Cabeçalho
15     cabecalho();
16     //Início do Programa
17     //Entrada com validação com mensagem de ERRO
18     printf("Digite quantos alunos a turma possui (maximo: 100): ");
19     scanf("%d",&qnt);
20     while((qnt>100)|| (qnt<0))
21     {
22         printf("ERRO! Digite quantos alunos a turma possui (maximo: 100): ");
23         scanf("%d",&qnt);
24     }
25     for(cont=0;cont<qnt;cont++) //Recebimento de 'qnt' notas
26     {
27         printf("Digite a nota do aluno #d: ",cont+1);
28         scanf("%f",&notas[cont]);
29         while((notas[cont]<0)|| (notas[cont]>10))
30         {
31             printf("ERRO! Digite a nota do aluno #d: ",cont+1);
32             scanf("%f",&notas[cont]);
33         }
34         soma+=notas[cont];
35     }
36     //Resultados:
37     media=soma/qnt;
38     printf("\nA média de notas da turma é: %.2f", media);
39     getch(); // Função que aguarda o recebimento de um caractere via teclado
40 }

```

Figura 10.4 – Código do programa em C: *respexe102.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite quantos alunos a turma possui (maximo: 100): 8
Digite a nota do aluno 1: 8.5
Digite a nota do aluno 2: 7.5
Digite a nota do aluno 3: 7.5
Digite a nota do aluno 4: 6
Digite a nota do aluno 5: 10
Digite a nota do aluno 6: 9.5
Digite a nota do aluno 7: 3
Digite a nota do aluno 8: 8

A media de notas da turma e: 7.50
```

Figura 10.5 – Resultado da execução do código do programa em C: *respexe102.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite quantos alunos a turma possui (maximo: 100): 10
Digite a nota do aluno 1: 1.5
Digite a nota do aluno 2: 5.5
Digite a nota do aluno 3: 9
Digite a nota do aluno 4: 8.5
Digite a nota do aluno 5: 10
Digite a nota do aluno 6: 4.5
Digite a nota do aluno 7: 3
Digite a nota do aluno 8: 10
Digite a nota do aluno 9: 9.5
Digite a nota do aluno 10: 9

A media de notas da turma e: 7.05
```

Figura 10.6 – Resultado da execução do código do programa em C: *respexe102.cpp*

Fonte: Elaboração própria

- 10.3 Escreva um programa que solicite as notas de quatro alunos ao usuário e apresente na tela a menor e a maior nota. Esse programa deve possuir um procedimento de cabeçalho e uma função para verificar a menor e a maior nota dos alunos possuindo um vetor de entrada NotAlunos[4].

Resposta: o código do programa é apresentado na Figura 10.7 e o resultado da execução desse código nas Figuras 10.8 e 10.9.

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabeçalho
5  void cabecalho(){
6      printf("Universidade de Brasília\n", 161);
7      printf("Disciplina: 113913 - ICC \n");
8      printf("Aluno(a):\n\n");
9  }
10 //função para verificar da maior nota
11 void verNotas(float NotAlunos[4]){
12     int cont;
13     float maior=0, menor=10;
14     //Verificação nota por nota
15     for(cont=0;cont<4;cont++)
16     {
17         if(NotAlunos[cont]>maior)
18             maior=NotAlunos[cont];
19         if(NotAlunos[cont]<menor)
20             menor=NotAlunos[cont];
21     }
22     //Resultados:
23     printf("\nA maior nota é: %.2f",maior);
24     printf("\nA menor nota é: %.2f",menor);
25 }
26 int main ()//Programa principal
27 // Declaração de variáveis:
28 float NotAlunos[4];
29 int cont;
30 //Chamada do procedimento Cabeçalho
31 cabecalho();
32 //Início do Programa
33 //Recebendo notas:
34 for(cont=0;cont<4;cont++)
35 {
36     //Validando notas
37     printf("Digite a nota do aluno %d: ",cont+1);
38     scanf("%f", &NotAlunos[cont]);
39     while((NotAlunos[cont]<0) || (NotAlunos[cont]>10))
40     {
41         printf("ERRO! Digite a nota do aluno %d: ",cont+1);
42         scanf("%f", &NotAlunos[cont]);
43     }
44 }
45 //Passando o vetor NotAlunos para a função verificar().
46 verNotas(NotAlunos);
47 getch(); // Função que aguarda o recebimento de um caractere via teclado
48 }
```

Figura 10.7 – Código do programa em C: *respexe103.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite a nota do aluno 1: 9.5
Digite a nota do aluno 2: 10
Digite a nota do aluno 3: 4.5
Digite a nota do aluno 4: 6

A maior nota e: 10.00
A menor nota e: 4.50
```

Figura 10.8 – Resultado da execução do código do programa em C: *respexe103.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite a nota do aluno 1: 3.5
Digite a nota do aluno 2: 7
Digite a nota do aluno 3: 8
Digite a nota do aluno 4: 8

A maior nota e: 8.00
A menor nota e: 3.50
```

Figura 10.9 – Resultado da execução do código do programa em C: *respexe103.cpp*

Fonte: Elaboração própria

- 10.4 Escreva um programa que solicite seis números do tipo inteiro ao usuário e os armazene em um vetor, depois o programa deverá apresentar na tela os números na ordem inversa do qual foram digitados.

Resposta: o código do programa é apresentado na Figura 10.10 e o resultado da execução desse código na Figura 10.11.

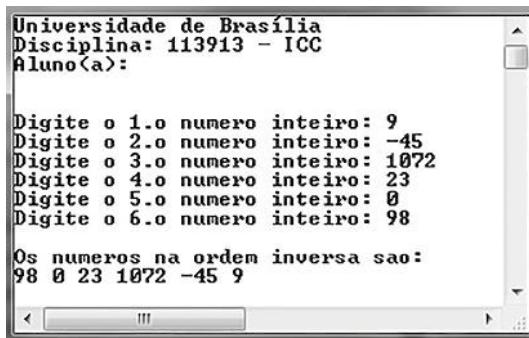
```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabeçalho
5  void cabecalho(){
6      printf("Universidade de Brasília\n", 161);
7      printf("Disciplina: 113913 - ICC \n");
8      printf("Aluno(a):\n\n");
9  }
10 int main (){ //Programa principal
11     // Declaração de variáveis:
12     int nros[6], cont;
13     //Chamada do procedimento Cabeçalho
14     cabecalho();
15     //Início do Programa
16     for(cont=0;cont<6;cont++)
17     {
18         printf("Digite o %d.o numero inteiro: ",cont+1);
19         scanf("%d",&nros[cont]);
20     }
21     printf("\nOs numeros na ordem inversa sao:\n");
22     //Imprimindo o resultado utilizando 'for' inversamente
23     for(cont=5;cont>=0;cont--)
24     {
25         //imprimindo resultados na tela
26         printf("%d ",nros[cont]);
27     }
28     getch(); // Função que aguarda o recebimento de um caractere via teclado
29 }

```

Figura 10.10 – Código do programa em C: *respexe104.cpp*

Fonte: Elaboração própria

Figura 10.11 – Resultado da execução do código do programa em C: *respexe104.cpp*

Fonte: Elaboração própria

- 10.5 Escreva um programa que leia doze números do tipo inteiro ao usuário. Separe esses números em pares e ímpares e os armazenem em dois outros vetores chamados vetpar e vetimpar. Em seguida, o programa dever apresentar os resultados na tela.

Resposta: o código do programa é apresentado na Figura 10.12 e a execução desse código nas Figuras 10.13 e 10.14.

```
1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho()
6 {
7     printf("Universidade de Brasília\n", 161);
8     printf("Disciplina: 113913 - ICC \n");
9     printf("Aluno(a):\n\n\n");
10 }
11 int main (){//Programa principal
12     // Declaração de variáveis:
13     int num[12], vetpar[12], vetimpar[12];
14     int cont, cont_par=0, cont_impar=0;
15     //Chamada do procedimento Cabeçalho
16     cabecalho();
17     //Início do Programa
18     printf("Digite doze numeros inteiros:\n");
19     // Receber os doze números via teclado
20     for (cont=0; cont<12; cont++)
21     {
22         printf("Digite o %d.o numero: ", cont+1);
23         scanf("%d", &num[cont]);
24         if (num[cont]&2==0)
25         {
26             vetpar[cont_par]=num[cont];
27             cont_par++;
28         }
29         else
30         {
31             vetimpar[cont_impar]=num[cont];
32             cont_impar++;
33         }
34     }
35     //Resultados:
36     printf("\nNumeros pares sao: \n");
```

```

36   for (cont=0; cont<cont_par; cont++)
37     printf("%d ", vetpar[cont]);
38
39   printf("\nNumeros impares:\n");
40   for (cont=0; cont<cont_impar; cont++)
41     printf("%d ", vetimpar[cont]);
42   getch(); // Função que aguarda o recebimento de um caractere via teclado
43 }

```

Figura 10.12 – Código do programa em C: *respexe105.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite doze numeros inteiros:
Digite o 1.o numero: 1
Digite o 2.o numero: 2
Digite o 3.o numero: 3
Digite o 4.o numero: 5
Digite o 5.o numero: 7
Digite o 6.o numero: 9
Digite o 7.o numero: 13
Digite o 8.o numero: 18
Digite o 9.o numero: 26
Digite o 10.o numero: 28
Digite o 11.o numero: 19
Digite o 12.o numero: 15

Numeros pares sao:
2 18 26 28
Numeros impares:
1 3 5 7 9 13 19 15

```

Figura 10.13 – Resultado da execução do código do programa em C: *respexe105.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Digite doze numeros inteiros:
Digite o 1.o numero: 9
Digite o 2.o numero: 6
Digite o 3.o numero: 7
Digite o 4.o numero: 5
Digite o 5.o numero: 3
Digite o 6.o numero: 4
Digite o 7.o numero: 1
Digite o 8.o numero: 2
Digite o 9.o numero: 8
Digite o 10.o numero: 0
Digite o 11.o numero: 11
Digite o 12.o numero: 12

Numeros pares sao:
6 4 2 8 0 12
Numeros impares:
9 7 5 3 1 11
```

Figura 10.14 – Resultado da execução do código do programa em C: *respexe105.cpp*

Fonte: Elaboração própria

- 10.6 Desenvolva um programa que leia dez números do tipo inteiro ao usuário. Armazene esses dez números em um vetor. Para os valores dos elementos inseridos nas posições pares desse vetor, calcule o somatório deles, para os demais calcule a subtração desses valores. Em seguida, o programa deverá apresentar na tela os resultados.

Resposta: o código do programa é apresentado na Figura 10.15 e a execução desse código nas Figuras 10.16 e 10.17.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 int main (){//Programa principal
11     // Declaração de variáveis:
12     int nros[10], cont, soma=0, sub=0;
13     //Chamada do procedimento Cabeçalho
14     cabecalho();
15     //Início do Programa
16     for(cont=0;cont<10;cont++)
17     {
18         printf("Digite o %d.o numero inteiro: ",cont+1);
19         scanf("%d",&nros[cont]);
20         //condição para subtração no caso ímpar e soma no caso par
21         if(cont%2!=0)
22             sub-=nros[cont];
23         else
24             soma+=nros[cont];
25     }
26     //Resultados:
27     printf("\nA soma dos numeros das posicoes pares e: %d", soma);
28     printf("\nA subtracao dos numeros das posicoes impares e: %d",sub);
29     getch(); // Função que aguarda o recebimento de um caractere via teclado
30 }
```

Figura 10.15 – Código do programa em C: *respexe106.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Digite o 1.o numero inteiro: 1
Digite o 2.o numero inteiro: 4
Digite o 3.o numero inteiro: 3
Digite o 4.o numero inteiro: 7
Digite o 5.o numero inteiro: 9
Digite o 6.o numero inteiro: 6
Digite o 7.o numero inteiro: 8
Digite o 8.o numero inteiro: 10
Digite o 9.o numero inteiro: 2
Digite o 10.o numero inteiro: 3

A soma dos numeros das posicoes pares e: 23
A subtracao dos numeros das posicoes impares e: -30

```

Figura 10.16 – Resultado da execução do código do programa em C: *respexe106.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o 1.o numero inteiro: 3
Digite o 2.o numero inteiro: 5
Digite o 3.o numero inteiro: 4
Digite o 4.o numero inteiro: 6
Digite o 5.o numero inteiro: 7
Digite o 6.o numero inteiro: 5
Digite o 7.o numero inteiro: 9
Digite o 8.o numero inteiro: 0
Digite o 9.o numero inteiro: 1
Digite o 10.o numero inteiro: 3

A soma dos numeros das posicoes pares e: 24
A subtracao dos numeros das posicoes impares e: -19
```

Figura 10.17 – Resultado da execução do código do programa em C: *respexe106.cpp*

Fonte: Elaboração própria

Capítulo 11

Matrizes

Matrizes: exercícios

11.1 Faça um programa para obter os valores de uma matriz 6x6 de números inteiros. Depois da leitura dos dados, o programa deve calcular a soma dos elementos da diagonal principal. Em seguida ele deve mostrar os valores da matriz e o da soma.

Resposta: o código do programa é apresentado na Figura 11.1 e o resultado da execução desse código na Figura 11.2.

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabeçalho
5  void cabeçalho()
6  {
7      printf("Universidade de Brasília\n", 161);
8      printf("Disciplina: 113913 - ICC \n");
9      printf("Aluno(a):\n\n");
10 }
11 int main ()//Programa principal
12 {
13     // Declaração de variáveis:
14     int matriz[6][6];// Declaração de variáveis onde matriz[6][6] receberá os 36 números
15     int soma = 0, contlinhas, contcolunas;    // e soma para acumular a soma
16     //Chamada do procedimento Cabeçalho
17     cabeçalho();
18     //Início do Programa
19     for(contlinhas=0 ; contlinhas<6 ; contlinhas++)
20     {
21         printf("\nInforme os elementos da %d.a linha: \n",contlinhas+1);
22         for(contcolunas = 0 ; contcolunas<6 ; contcolunas++)
23         {
24             printf("Digite o elemento da %d.a coluna: ",contcolunas+1);
25             scanf("%d",&matriz[contlinhas][contcolunas]);
26         }
27         printf("\nApresentação da Matriz:\n");
28         //Apresentação dos resultados
29         for(contlinhas=0 ; contlinhas<6 ; contlinhas++)
30         {
31             for(contcolunas = 0 ; contcolunas<6 ; contcolunas++)
32             {
33                 printf("%d\t", matriz[contlinhas][contcolunas]);
34                 if(contlinhas==contcolunas)
35                     soma = soma + matriz[contlinhas][contcolunas];
36             }
37             printf("\n");
38         }
39         printf("Soma da diagonal principal: %d",soma);
40         getch(); // Função que aguarda o recebimento de um caractere via teclado
}

```

Figura 11.1 – Código do programa em C: *respexe111.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Informe os elementos da 1.a linha:
Digite o elemento da 1.a coluna: 1
Digite o elemento da 2.a coluna: 2
Digite o elemento da 3.a coluna: 3
Digite o elemento da 4.a coluna: 4
Digite o elemento da 5.a coluna: 5
Digite o elemento da 6.a coluna: 6

Informe os elementos da 2.a linha:
Digite o elemento da 1.a coluna: 6
Digite o elemento da 2.a coluna: 5
Digite o elemento da 3.a coluna: 4
Digite o elemento da 4.a coluna: 3
Digite o elemento da 5.a coluna: 2
Digite o elemento da 6.a coluna: 1

Informe os elementos da 3.a linha:
Digite o elemento da 1.a coluna: 11
Digite o elemento da 2.a coluna: 22
Digite o elemento da 3.a coluna: 33
Digite o elemento da 4.a coluna: 44
Digite o elemento da 5.a coluna: 55
Digite o elemento da 6.a coluna: 66

Informe os elementos da 4.a linha:
Digite o elemento da 1.a coluna: 66
Digite o elemento da 2.a coluna: 55
Digite o elemento da 3.a coluna: 44
Digite o elemento da 4.a coluna: 33
Digite o elemento da 5.a coluna: 22
Digite o elemento da 6.a coluna: 11

Informe os elementos da 5.a linha:
Digite o elemento da 1.a coluna: 0
Digite o elemento da 2.a coluna: 9
Digite o elemento da 3.a coluna: 8
Digite o elemento da 4.a coluna: 7
Digite o elemento da 5.a coluna: 6
Digite o elemento da 6.a coluna: 5

Informe os elementos da 6.a linha:
Digite o elemento da 1.a coluna: 4
Digite o elemento da 2.a coluna: 3
Digite o elemento da 3.a coluna: 2
Digite o elemento da 4.a coluna: 1
Digite o elemento da 5.a coluna: 1
Digite o elemento da 6.a coluna: 2

Apresentacao da Matriz:
1   2   3   4   5   6
6   5   4   3   2   1
11  22  33  44  55  66
66  55  44  33  22  11
0   9   8   7   6   5
4   3   2   1   1   2
Soma da diagonal principal: 80
```

Figura 11.2 – Resultado da execução do código do programa em C: *respexe111.cpp*

Fonte: Elaboração própria

- 11.2 Desenvolva um programa para ler os dados de uma matriz para armazenar três notas de dez alunos. Em seguida apresente a menor nota da prova de cada aluno. Desenvolva uma função para obter a menor nota de cada aluno.

Resposta: o código do programa é apresentado na Figura 11.3 e o resultado da execução desse código na Figura 11.4.

```

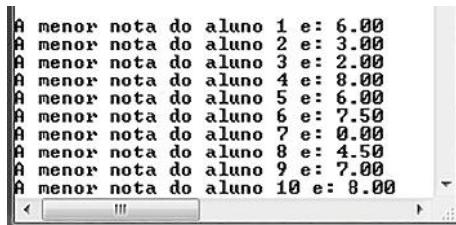
1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabeçalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 //Função que irá retornar a menor nota
11 float menorNota(float notas[3]){
12     float menor_nota = 10;
13     int cont;
14     for(cont=0; cont<3; cont++){
15         if(notas[cont] < menor_nota)
16             menor_nota = notas[cont];
17     }
18     return menor_nota;
19 }
20 int main (){//Programa principal
21     // Declaração de variáveis:
22     float notas[10][3], menores[10];
23     int contl, contc;
24     //Chamada do procedimento Cabeçalho
25     cabecalho();
26     //Início do Programa
27     // Percorrendo a matriz para receber dados via teclado
28     for(contl=0; contl<10; contl++){
29         printf("Digite as notas do %d.o aluno: \n", contl+1);
30         for(contc=0; contc<3; contc++){
31             printf("Digite a %d.a nota: ", contc+1);
32             scanf("%f", &notas[contl][contc]);
33             while(notas[contl][contc]<0 || notas[contl][contc]>10){
34                 printf("Erro! Digite corretamente a %d.a nota do %d.o aluno: ", contc+1, contl+1);
35                 scanf("%f", &notas[contl][contc]);
36             }
37         }
38         menores[contl] = menorNota(notas[contl]); //Passando uma linha para a função
39         printf("\n");
40     }
41     // Apresentação de resultados
42 }
43 
```

```
44     for(contl=0; contl<10; contl++)  
45         printf("A menor nota do aluno %d é: %.2f\n",contl+1,menores[contl]);  
46     getch();// Função que aguarda o recebimento de um caractere via teclado  
47 }
```

Figura 11.3 – Código do programa em C: *respexe112.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília  
Disciplina: 113913 - ICC  
Aluno<a>:  
  
Digite as notas do 1.o aluno:  
Digite a 1.a nota: 9.5  
Digite a 2.a nota: 6.5  
Digite a 3.a nota: 6  
  
Digite as notas do 2.o aluno:  
Digite a 1.a nota: 7  
Digite a 2.a nota: 3  
Digite a 3.a nota: 8.5  
  
Digite as notas do 3.o aluno:  
Digite a 1.a nota: 2  
Digite a 2.a nota: 4  
Digite a 3.a nota: 4.5  
  
Digite as notas do 4.o aluno:  
Digite a 1.a nota: 9.5  
Digite a 2.a nota: 10  
Digite a 3.a nota: 8  
  
Digite as notas do 5.o aluno:  
Digite a 1.a nota: 6  
Digite a 2.a nota: 7  
Digite a 3.a nota: 7  
  
Digite as notas do 6.o aluno:  
Digite a 1.a nota: 10  
Digite a 2.a nota: 8.5  
Digite a 3.a nota: 7.5  
  
Digite as notas do 7.o aluno:  
Digite a 1.a nota: 7.5  
Digite a 2.a nota: 1  
Digite a 3.a nota: 0  
  
Digite as notas do 8.o aluno:  
Digite a 1.a nota: 4.5  
Digite a 2.a nota: 5  
Digite a 3.a nota: 6  
  
Digite as notas do 9.o aluno:  
Digite a 1.a nota: 7  
Digite a 2.a nota: 7  
Digite a 3.a nota: 8  
  
Digite as notas do 10.o aluno:  
Digite a 1.a nota: 8  
Digite a 2.a nota: 8  
Digite a 3.a nota: 8
```



```
A menor nota do aluno 1 e: 6.00
A menor nota do aluno 2 e: 3.00
A menor nota do aluno 3 e: 2.00
A menor nota do aluno 4 e: 8.00
A menor nota do aluno 5 e: 6.00
A menor nota do aluno 6 e: 7.50
A menor nota do aluno 7 e: 0.00
A menor nota do aluno 8 e: 4.50
A menor nota do aluno 9 e: 7.00
A menor nota do aluno 10 e: 8.00
```

Figura 11.4 – Resultado da execução do código do programa em C: *respexe112.cpp*

Fonte: Elaboração própria

11.3 Desenvolva um programa que possua uma matriz notaprova para armazenar quatro notas de cinco alunos e depois apresentem na tela essas notas. Esse programa dever possuir validação de dados de entrada. Utilizar duas estruturas de repetição uma para manusear a coluna da matriz e outra para manusear as linhas. O resultado do programa deve ser conforme as figuras a seguir, a primeira figura representa a obtenção dos dados e a segunda à apresentação dos dados na tela.

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite as quatro notas do 1.o aluno:
Digite 1.a nota: 8
Digite 2.a nota: 9.5
Digite 3.a nota: 16
Voce digitou uma nota maior do que dez
Digite novamente a 3.o nota: 9.5
Digite 4.a nota: 7

Digite as quatro notas do 2.o aluno:
Digite 1.a nota: 4.5
Digite 2.a nota: -2
Voce digitou uma nota menor do que zer
Digite novamente a 2.o nota: 2
Digite 3.a nota: 33
Voce digitou uma nota maior do que dez
Digite novamente a 3.o nota: 3
Digite 4.a nota: 3

Digite as quatro notas do 3.o aluno:
Digite 1.a nota: 4.5
Digite 2.a nota: 5
Digite 3.a nota: 6
Digite 4.a nota: 6

Digite as quatro notas do 4.o aluno:
Digite 1.a nota: 9
Digite 2.a nota: 7.5
Digite 3.a nota: ?
Digite 4.a nota: 10

Digite as quatro notas do 5.o aluno:
Digite 1.a nota: 10
Digite 2.a nota: 6
Digite 3.a nota: 6.5
Digite 4.a nota: 8
```

Figura 11.5 – Trecho do resultado da execução do código do programa em C a ser elaborado, que obtém as notas dos estudantes: *respexe113.cpp*

Fonte: Elaboração própria

```
As notas do 1.o aluno sao:  
1.a nota: 8.00  
2.a nota: 9.50  
3.a nota: 9.50  
4.a nota: 7.00  
  
As notas do 2.o aluno sao:  
1.a nota: 4.50  
2.a nota: 2.00  
3.a nota: 3.00  
4.a nota: 3.00  
  
As notas do 3.o aluno sao:  
1.a nota: 4.50  
2.a nota: 5.00  
3.a nota: 6.00  
4.a nota: 6.00  
  
As notas do 4.o aluno sao:  
1.a nota: 9.00  
2.a nota: 7.50  
3.a nota: 7.00  
4.a nota: 10.00  
  
As notas do 5.o aluno sao:  
1.a nota: 10.00  
2.a nota: 6.00  
3.a nota: 6.50  
4.a nota: 8.00
```

Figura 11.6 – Trecho do resultado da execução do código do programa em C a ser elaborado, que apresenta as notas dos estudantes: *respexe113.cpp*

Fonte: Elaboração própria

Resposta: o código do programa é apresentado na Figura 11.7.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 // Função para retornar a nota validada
11 float validaNota(float nota, int cont) {
12     while(nota<0 || nota>10)
13     {
14         if(nota<0)
15         {
16             printf("Voce digitou uma nota menor do que zero: ");
17             printf("\nDigite novamente a %d.o nota: ", cont+1);
18             scanf("%f", &nota);
19         }
20         if(nota>10)
21         {
22             printf("Voce digitou uma nota maior do que dez: ");
23             printf("\nDigite novamente a %d.o nota: ", cont+1);
24             scanf("%f", &nota);
25         }
26     }
27     return nota;
28 }
29 int main (){//Programa principal
30     // Declaração de variáveis:
31     float notaprova[5][4];
32     int cont1, contc;
33     //Chamada do procedimento Cabeçalho
34     cabecalho();
35     //Início do Programa
36     // Receber dados via teclado realizando a validação
37     for(cont1=0; cont1<5; cont1++)
38     {
39         printf("\n\nDigite as quatro notas do %d.o aluno: \n",cont1+1);
40         for(contc=0; contc<4; contc++)
41         {
42             printf("Digite %d.a nota: ", contc+1);
43             scanf("%f", &notaprova[cont1][contc]);
44             //Validação pela função da nota pelo de determinado índice 'contc'
45             notaprova[cont1][contc]= validaNota(notaprova[cont1][contc], contc);
46         }
47     }
48     // Apresentando resultados na tela
49     printf("\n");
50     for(cont1=0; cont1<5; cont1++)
51     {
52         printf("\nAs notas do %d.o aluno são:\n", cont1+1);
53         for(contc=0; contc<4; contc++)
54             printf("%d.a nota: %.2f\n", contc+1,notaprova[cont1][contc]);
55     }
56     getch(); // Função que aguarda o recebimento de um caractere via teclado
57 }
```

Figura 11.7 – Código do programa em C: *respexe113.cpp*

Fonte: Elaboração própria

- 11.4 Desenvolva um programa que possua uma matriz de 4X8 de números inteiros e some cada uma das linhas da matriz, guardando o resultado da soma em um vetor somalinha. A seguir, apresente os valores da matriz e do vetor.

Resposta: o código do programa é apresentado na Figura 11.8 e o resultado da execução desse código na Figura 11.9.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho()
6 {
7     printf("Universidade de Brasília\n", 161);
8     printf("Disciplina: 113913 - ICC \n");
9     printf("Aluno(a):\n\n");
10 }
11 int main (){//Programa principal
12     // Declaração de variáveis:
13     int matriz[4][8], somalinha[4] = {0,0,0,0}; //Inicialização de vetor
14     int contl, contc;
15     //Chamada do procedimento Cabeçalho
16     cabecalho();
17     //Início do Programa
18     for(contl=0; contl<4; contl++)
19     {
20         printf("Digite os elementos da %d.a linha: \n",contl+1);
21         for(contc=0; contc<8; contc++)
22         {
23             printf("Digite o elemento da %d.a coluna: ", contc+1);
24             scanf("%d",&matriz[contl][contc]);
25         }
26         printf("\n");
27         // Realizando Soma da linha
28         for(contl=0; contl<4; contl++)
29         {
30             for(contc=0; contc<8; contc++)
31                 somalinha[contl]=somalinha[contl] + matriz[contl][contc];
32             //Apresentando resultados
33             for(contl=0; contl<4; contl++)
34             {
35                 printf("\n");
36                 for(contc=0; contc<8; contc++)
37                     printf("%d ",matriz[contl][contc]);
38             }
39             printf("\n\n");
40             for(contl=0; contl<4; contl++)
41                 printf("Soma da linha %d: %d\n",contl+1,somalinha[contl]);
42             getch(); // Função que aguarda o recebimento de um caractere via teclado
        }
    }
}

```

Figura 11.8 – Código do programa em C: *respexe114.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite os elementos da 1.a linha:
Digite o elemento da 1.a coluna: 1
Digite o elemento da 2.a coluna: 3
Digite o elemento da 3.a coluna: 5
Digite o elemento da 4.a coluna: 7
Digite o elemento da 5.a coluna: 9
Digite o elemento da 6.a coluna: 1
Digite o elemento da 7.a coluna: 3
Digite o elemento da 8.a coluna: 6

Digite os elementos da 2.a linha:
Digite o elemento da 1.a coluna: 6
Digite o elemento da 2.a coluna: 4
Digite o elemento da 3.a coluna: 2
Digite o elemento da 4.a coluna: 4
Digite o elemento da 5.a coluna: 2
Digite o elemento da 6.a coluna: 8
Digite o elemento da 7.a coluna: 6
Digite o elemento da 8.a coluna: 4

Digite os elementos da 3.a linha:
Digite o elemento da 1.a coluna: 1
Digite o elemento da 2.a coluna: 2
Digite o elemento da 3.a coluna: 3
Digite o elemento da 4.a coluna: 4
Digite o elemento da 5.a coluna: 5
Digite o elemento da 6.a coluna: 6
Digite o elemento da 7.a coluna: 7
Digite o elemento da 8.a coluna: 8

Digite os elementos da 4.a linha:
Digite o elemento da 1.a coluna: 9
Digite o elemento da 2.a coluna: 8
Digite o elemento da 3.a coluna: 7
Digite o elemento da 4.a coluna: 6
Digite o elemento da 5.a coluna: 5
Digite o elemento da 6.a coluna: 4
Digite o elemento da 7.a coluna: 3
Digite o elemento da 8.a coluna: 2

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

Soma da linha 1: 35
Soma da linha 2: 36
Soma da linha 3: 36
Soma da linha 4: 44
```

Figura 11.9 – Resultado da execução do código do programa em C: *respexe114.cpp*

Fonte: Elaboração própria

- 11.5 Faça um programa para ler uma matriz com três linhas e quatro colunas; cada linha se refere aos dados de um aluno; cada coluna se refere à nota de um bimestre. Calcular a média de cada um dos alunos e dizer se o aluno foi aprovado ou não (a média é igual a sete).

Resposta: o código do programa é apresentado na Figura 11.10 e o resultado da execução desse código nas Figuras 11.11 e 11.12.

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabecalho
5  void cabecalho()
6  {
7      printf("Universidade de Brasília\n", 161);
8      printf("Disciplina: 113913 - ICC \n");
9      printf("Aluno(a):\n\n");
10 }
11 //Função que recebe as notas de um aluno e verifica se foi aprovado ou não
12 //Funções bool retornam valores lógicos (verdadeiro ou falso)
13 bool verificaAprovacao(float resultados[])
14 {
15     float soma = 0;
16     int cont;
17     for(cont=0; cont<4; cont++)
18     {
19         soma += resultados[cont];
20     }
21     return ((soma/4.0) >= 7.0);
22 }
23 int main ()//Programa principal
24 {
25     // Declaração de variáveis:
26     float notas[3][4];
27     int cont1, contc;
28     //Chamada do procedimento Cabecalho
29     cabecalho();
30     //Início do Programa
31     for(cont1=0; cont1<3; cont1++)
32     {
33         printf("Digite as notas do %d.o aluno: ", cont1+1);
34         for(contc=0; contc<4; contc++)
35         {
36             printf("Digite a %d.a nota: ", contc+1);
37             scanf("%f",&notas[cont1][contc]);
38             while(notas[cont1][contc]<0 || notas[cont1][contc]>10)
39             {
40                 printf("Erro! Digite novamente a %d.a nota do %d.o aluno(a): ", contc+1, cont1+1);
41                 scanf("%f",&notas[cont1][contc]);
42             }
43         }
44         //para cada aluno, verificar as notas e seu status (aprovado ou reprovado)
45         if(verificaAprovacao(notas[cont1])==true)
46         {
47             printf("\nO %d.o aluno esta aprovado.", cont1+1);
48         }
49     }
50     getch(); // Função que aguarda o recebimento de um caractere via teclado
51 }
```

Figura 11.10 – Código do programa em C: *respexe115.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite as notas do 1.o aluno:
Digite a 1.a nota: 1
Digite a 2.a nota: 3.5
Digite a 3.a nota: 4
Digite a 4.a nota: 7.5

Digite as notas do 2.o aluno:
Digite a 1.a nota: 6
Digite a 2.a nota: 7.5
Digite a 3.a nota: 9
Digite a 4.a nota: 5.5

Digite as notas do 3.o aluno:
Digite a 1.a nota: 7
Digite a 2.a nota: 6
Digite a 3.a nota: 10
Digite a 4.a nota: 10

O 1.o aluno está reprovado
O 2.o aluno está aprovado.
O 3.o aluno está aprovado.
```

Figura 11.11 – Resultado da execução do código do programa em C: *respexe115.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite as notas do 1.o aluno:
Digite a 1.a nota: 2
Digite a 2.a nota: 5
Digite a 3.a nota: 4
Digite a 4.a nota: 7

Digite as notas do 2.o aluno:
Digite a 1.a nota: 7
Digite a 2.a nota: 9
Digite a 3.a nota: 2
Digite a 4.a nota: 10

Digite as notas do 3.o aluno:
Digite a 1.a nota: 10
Digite a 2.a nota: 9
Digite a 3.a nota: 8
Digite a 4.a nota: 9

O 1.o aluno está reprovado
O 2.o aluno está aprovado.
O 3.o aluno está aprovado.
```

Figura 11.12 – Resultado da execução do código do programa em C: *respexe115.cpp*

Fonte: Elaboração própria

11.6 Faça um programa que leia uma matriz N por N, para a qual a dimensão deve ser fornecida pelo usuário. O programa deve subtrair a soma da diagonal principal pela soma da diagonal secundária e exibir o resultado.

Resposta: o código do programa é apresentado na Figura 11.13 e o resultado da execução desse código na Figura 11.14.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho()
6 {
7     printf("Universidade de Brasília\n", 161);
8     printf("Disciplina: 113913 - ICC \n");
9     printf("Aluno(a):\n\n");
10 }
11 int main ()//Programa principal
12 {
13     // Declaração de variáveis:
14     int dim, contl, contc;
15     //Chamada do procedimento Cabeçalho
16     cabecalho();
17     //Início do Programa
18     printf("Digite a dimensão da matriz quadrada: ");
19     scanf("%d",&dim);
20     printf("\n");
21     //Segunda fase de declaração de variáveis:
22     int matriz[dim][dim], diag_principal=0, diag_sec=0;
23     for(contl=0; contl<dim; contl++)
24     {
25         printf("Digite os valores da %d.a linha:\n",contl+1);
26         for(contc=0; contc<dim; contc++)
27         {
28             printf("Digite o valor da %d.a coluna: ", contc+1);
29             scanf("%d",&matriz[contl][contc]);
30             // Somar a diagonal principal
31             if(contl==contc)
32                 diag_principal += matriz[contl][contc];
33         }
34         printf("\n");
35     }
36     //Calculando a partir da relação entre linha e coluna para a diagonal
37     for(contl=0; contl<dim; contl++)
38     {
39         diag_sec += matriz[contl][dim-contl-1];
40     }
41     //Resultados:
42     printf("\nMatriz:");
43     for(contl=0; contl<dim; contl++)
44     {
45         printf("\n");
46         for(contc=0; contc<dim; contc++)
47             printf("%d\t",matriz[contl][contc]);
48     }
49     printf("\nA soma da diagonal principal menos a soma da diagonal secundária é %d\n", diag_principal - diag_sec);
50     getch(); // Função que guarda o recebimento de um caractere via teclado
51 }
```

Figura 11.13 – Código do programa em C: *respexe116.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite a dimensão da matriz quadrada: 3  
  
Digite os valores da 1.a linha:  
Digite o valor da 1.a coluna: 10  
Digite o valor da 2.a coluna: 11  
Digite o valor da 3.a coluna: 19  
  
Digite os valores da 2.a linha:  
Digite o valor da 1.a coluna: 18  
Digite o valor da 2.a coluna: 12  
Digite o valor da 3.a coluna: 13  
  
Digite os valores da 3.a linha:  
Digite o valor da 1.a coluna: 17  
Digite o valor da 2.a coluna: 16  
Digite o valor da 3.a coluna: 15  
  
Matriz:  
10      11      19  
18      12      13  
17      16      15  
  
A soma da diagonal principal menos a soma da diagonal secundária é -11
```

Figura 11.14 – Resultado da execução do código do programa em C: *respexe116.cpp*

Fonte: Elaboração própria

Capítulo 12

Strings

Strings

12.1 Desenvolva um programa para ler uma string de no máximo 100 caracteres e informe o número que cada vogal se repete nessa frase.

Resposta: o código do programa é apresentado na Figura 12.1 e o resultado do código da execução desse código na Figura 12.2.

```

1 #include<string.h>/Biblioteca especifica para funções strings
2 #include<stdio.h> /Biblioteca de entrada/saída por periféricos padrão
3 #include<conio.h> //Biblioteca para manipulação de caracteres
4 #include<stdlib.h>/Biblioteca padrão para uso de funções do sistema
5 //Procedimento Cabecalho
6 void cabecalho(){
7     printf("Universidade de Brasília\n", 161);
8     printf("Disciplina: 113913 - ICC \n");
9     printf("Aluno(a):\n\n");
10 }
11 // Função para contagem de caracteres de uma String
12 int strlen1(char *auxstring){
13     int cont=0;
14     while(auxstring[cont] != '\0')
15     {
16         cont++;
17     }
18     int main ()//Programa principal
19     // Declaração de variáveis:
20     char frase[100], opcao;
21     int tamanho, auxcont, contaA, contaE, contaI, contaO, contaU;
22     do
23     {
24         contaA=0, contaE=0, contaI=0, contaO=0, contaU=0; //Inicializando variáveis
25         //Chamada do procedimento Cabeçalho
26         cabecalho();
27         //Início do Programa
28         printf("Digite uma frase: ");
29         gets(frase); //Função que recebe uma string
30         tamanho = strlen1(frase);
31         puts(frase);
32         while((frase[0] == '\0') || tamanho > 100) // validação para tamanho de frase
33         {
34             puts("Erro! Digite uma frase: ");
35             gets(frase); //Função que recebe uma string

```

```

36     tamanho = strlen(frase);
37 }
38 for(auxcont=0;auxcont<tamanho;auxcont++)
39 {
40     //Estrutura condicional composta para contagem de número de vogais
41     if(frase[auxcont]=='a' || frase[auxcont]=='A')
42         contaA++;
43     else if(frase[auxcont]=='e' || frase[auxcont]=='E')
44         contaE++;
45     else if(frase[auxcont]=='i' || frase[auxcont]=='I')
46         contaI++;
47     else if(frase[auxcont] == 'o' || frase[auxcont] == 'O')
48         contaO++;
49     else if(frase[auxcont] == 'u' || frase[auxcont] == 'U')
50         contaU++;
51 }
52 //Resultados:
53 printf("\nA frase %c%c possui a seguinte quantidade de vogais:\n",34,frase,34);
54 printf("\ncA%c: %d",34,34,contaA);
55 printf("\ncE%c: %d",34,34,contaE);
56 printf("\ncI%c: %d",34,34,contaI);
57 printf("\ncO%c: %d",34,34,contaO);
58 printf("\ncU%c: %d",34,34,contaU);
59 printf("\n\nDeseja executar novamente (S/N): ");
60 scanf("%c", &opcao);
61 fflush(stdin); //Função para limpar o buffer do teclado
62 while(opcao!='s' && opcao!='S' && opcao!='n' && opcao!='N')
63 {
64     printf("\n\nErro! Deseja executar novamente (S/N): ");
65     scanf("%c", &opcao);
66     fflush(stdin); //Função para limpar o buffer do teclado
67 }
68 system("cls");//Limpando a tela
69 }while(opcao=='s' || opcao=='S');
70 }

```

Figura 12.1 – Código do programa em C: *respexe121.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite uma frase: Utilizando a apostila
Utilizando a apostila

A frase "Utilizando a apostila" possui a seguinte quantidade de vogais:

"A": 4
"E": 0
"I": 3
"O": 2
"U": 1

Deseja executar novamente (S/N):

```

Figura 12.2 – Resultado da execução do código do programa em C: *respexe121.cpp*

Fonte: Elaboração própria

- 12.2 Desenvolva um programa que leia duas strings de tamanho máximo de 60 caracteres e informe se essas duas frases são iguais.

Resposta: o código do programa é apresentado na Figura 12.3 e o resultado da execução desse código nas Figuras 12.4 e 12.5.

```

1  #include<string.h>/>Biblioteca especifica para funções strings
2  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
3  #include<conio.h> //Biblioteca para manipulação de caracteres
4  #include<stdlib.h>/>Biblioteca padrão para uso de funções do sistema
5  //Procedimento Cabeçalho
6  void cabecalho()
7  {
8      printf("Universidade de Brasília\n", 161);
9      printf("Disciplina: 113913 - ICC \n");
10     printf("Aluno(a):\n\n\n");
11 }
12 int main (){//Programa principal
13     // Declaração de variáveis:
14     char frase1[60], frase2[60], opcao;
15     int cont, tamanho1, tamanho2, tamanhoFrase;
16     do
17     {
18         tamanhoFrase=0; //Inicializando variáveis
19         //Chamada do procedimento Cabeçalho
20         cabecalho();
21         //Início do Programa
22         printf("Digite a primeira frase: ");
23         gets(frase1);
24         printf("\nDigite a segunda frase: ");
25         gets(frase2);
26         /*strlen é a mesma função implementada no exercício anterior "strlen"
27         contudo a própria biblioteca string possui essa função específica de
28         contagem do tamanho de um string*/
29         tamanho1= strlen(frase1);
30         tamanho2= strlen(frase2);
31         // Verifica se as frases são iguais caractere por caractere
32         if(tamanho1 == tamanho2)
33         {
34             for(cont=0; cont<tamanho1; cont++)
35             {
36                 if(frase1[cont] == frase2[cont])
37                     tamanhoFrase++;
38                 else
39                     break; // Encerra o laço 'for'
40             }
41             // Comparação e apresentação de resultados
42             if(tamanho2 == tamanhoFrase)
43             {
44                 printf("\nFrases são iguais:\n");
45                 puts(frase1);
46             }
47         }
48     }
49 }
```

Figura 12.3 – Código do programa em C: *respexe122.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite a primeira frase: Aprendendo à estudar
Digite a segunda frase: Aprendendo à estudar
Frases são iguais:  
Aprendendo à estudar
Deseja executar novamente (S/N): S
```

Figura 12.4 – Resultado da execução do código do programa em C: *respexe122.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite a primeira frase: O céu é maravilhoso
Digite a segunda frase: As estrelas brilham intensamente
Frases são diferentes:  
O céu é maravilhoso
As estrelas brilham intensamente
Deseja executar novamente (S/N):
```

Figura 12.5 – Resultado da execução do código do programa em C: *respexe122.cpp*

Fonte: Elaboração própria

12.3 Desenvolva um programa que leia uma palavra de no máximo 60 caracteres e verifique se ela é um palíndromo. Uma palavra é palíndromo quando ela pode ser lida da mesma forma da esquerda para direita e da direita para esquerda.

Resposta: o código do programa é apresentado na Figura 12.6 e a execução desse código nas Figuras 12.7 e 12.8.

```

1  #include<string.h> //Biblioteca específica para funções strings
2  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
3  #include<conio.h> //Biblioteca para manipulação de caracteres
4  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
5  //Procedimento Cabeçalho
6  void cabecalho(){
7      printf("Universidade de Brasília\n", 161);
8      printf("Disciplina: 113913 - ICC \n");
9      printf("Aluno(a):\n\n");
10 }
11 int main (){//Programa principal
12     // Declaração de variáveis:
13     char palavra[60], opcao;
14     int cont1, cont2, tamanho, resposta=0;
15     /* cont1 percorrerá de traz pra frente e cont2
16     percorrerá normalmente mantendo-se as mesmas proporções.
17     tamanho recebe o tamanho da palavra e resposta é incrementada
18     toda vez que uma letra é identificada na posição oposta da string*/
19     do
20     {
21         resposta=0; // Inicialização de variáveis
22         //Chamada do procedimento Cabeçalho
23         cabecalho();
24         //Início do Programa
25         printf("Digite uma palavra: ");
26         gets(palavra);
27         tamanho=strlen(palavra); //Função string para contar o tamanho de um string
28         cont1=tamanho; // Atribuição para manipulação
29         for(cont2=0; cont2<tamanho; cont2++)
30         {
31             if(palavra[cont2]==palavra[cont1-1])
32                 resposta++;
33             cont1--;
34         }
35         //Comparações e resultados:
36         if(resposta==tamanho)
37             printf("\nA palavra %c%c%c é um palíndromo.",34,palavra,34,130,161);
38         else
39             printf("\nA palavra %c%c%c não é um palíndromo.",34,palavra,34,130,161);
40         printf("\nDeseja executar novamente (S/N): ");
41         scanf("%c", &opcao);
42         fflush(stdin); //Função para limpar o buffer do teclado
43         while(opcao!='s' && opcao!='S' && opcao!='n' && opcao!='N')
44         {
45             printf("\nErro! Deseja executar novamente (S/N): ");
46             scanf("%c", &opcao);
47             fflush(stdin); //Função para limpar o buffer do teclado
48         }
49         system("cls");//Limpando a tela
50     }while(opcao=='s' || opcao=='S');
51 }
52 }
```

Figura 12.6 – Código do programa em C: *respexe123.cpp*

Fonte: Elaboração própria

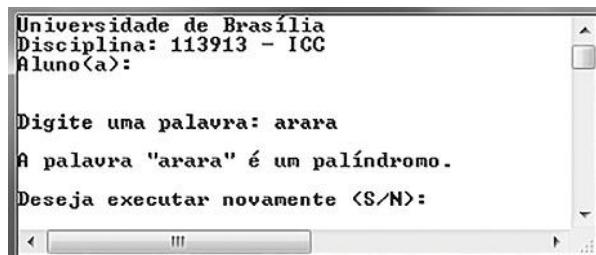


Figura 12.7 – Resultado da execução do código do programa em C: *respexe123.cpp*

Fonte: Elaboração própria

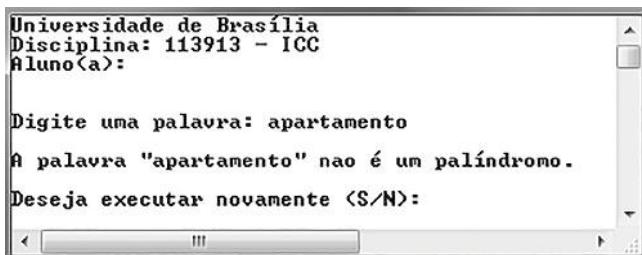


Figura 12.8 – Resultado da execução do código programa em C: *respexe123.cpp*

Fonte: Elaboração própria

- 12.4 Desenvolva um programa para ler uma palavra de no máximo 60 caracteres e faça o programa escrever essa palavra de trás para frente.

Resposta: o código do programa é apresentado na Figura 12.9 e o resultado da execução desse código nas Figuras 12.10 e 12.11.

```

1  #include<string.h> //Biblioteca específica para funções strings
2  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
3  #include<conio.h> //Biblioteca para manipulação de caracteres
4  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
5  //Procedimento Cabecalho
6  void cabecalho(){
7      printf("Universidade de Brasília\n", 161);
8      printf("Disciplina: 113913 - ICC \n");
9      printf("Aluno(a):\n\n");
10 }
11 int main (){//Programa principal
12     // Declaração de variáveis:
13     char palavra[60], opcao;
14     int tamanho, cont;
15     do
16     {
17         //Chamada do procedimento Cabecalho
18         cabecalho();
19         //Início do Programa
20         // Recebendo informações via teclado e interagindo com o usuário
21         printf("Digite uma palavra: ");
22         gets(palavra);
23         while(palavra[0]=='\0' || strlen(palavra)>60)
24         {
25             puts("Erro! Digite uma palavra válida: ");
26             gets(palavra);
27         }
28         tamanho = strlen(palavra);
29         // Imprimindo ao contrário o resultado:
30         printf("\nA palavra %c%c%c invertida %c: ",34,palavra,34,130);
31         for(cont=tamanho; cont>=0;cont--)
32             printf("%c",palavra[cont]);
33         printf("\n\nDeseja executar novamente (S/N): ");
34         scanf("%c", &opcao);
35         fflush(stdin); //Função para limpar o buffer do teclado
36         while(opcao!='s' && opcao!='S' && opcao!='n' && opcao!='N')
37         {
38             printf("\n\nErro! Deseja executar novamente (S/N): ");
39             scanf("%c", &opcao);
40             fflush(stdin); //Função para limpar o buffer do teclado
41         }
42         system("cls");//Limpa a tela
43     }while(opcao=='s' || opcao=='S');
44 }

```

Figura 12.9 – Código do programa em C: *respexe124.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite uma palavra: programação
A palavra "programação" invertida é: oãçamargorp
Deseja executar novamente <S/N>:  
!!!
```

Figura 12.10 – Resultado da execução do código do programa em C: *respexe124.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite uma palavra: informática
A palavra "informática" invertida é: acitámrnofni
Deseja executar novamente <S/N>:  
!!!
```

Figura 12.11 – Resultado da execução do código do programa em C: *respexe124.cpp*

Fonte: Elaboração própria

12.5 Escreva um programa em C que receba uma frase qualquer fornecida pelo usuário, e que calcule e mostre quantas palavras essa frase possui. A frase deve possuir no máximo 50 caracteres.

Resposta: o código do programa é apresentado na Figura 12.12 e o resultado da execução desse código nas Figuras 12.13 e 12.14.

```

1  #include<string.h> //Biblioteca específica para funções strings
2  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
3  #include<conio.h> //Biblioteca para manipulação de caracteres
4  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
5  //Procedimento Cabecalho
6  void cabecalho(){
7      printf("Universidade de Brasília\n", 161);
8      printf("Disciplina: 113913 - ICC \n");
9      printf("Aluno(a):\n\n");
10 }
11 int main (){//Programa principal
12     // Declaração de variáveis:
13     char frase[50], opcao;
14     int cont, auxcont;
15     do
16     {
17         auxcont=1; // Inicialização de Variáveis
18         /*auxcont começa em um pois ele conta quantos
19         espaços foram digitados, ou seja, quando se digita o primeiro espaço
20         já se tem uma palavra digitada*/
21         //Chamada do procedimento Cabeçalho
22         cabecalho();
23         //Início do Programa
24         printf("Digite uma frase: ");
25         gets(frase);
26         for (cont =0; cont< 50; cont++)
27             if(frase[cont]==' ')
28                 auxcont++;
29         //Resultado:
30         printf("\nA frase %s possui %d palavras.",frase, auxcont);
31         printf("\nDeseja executar novamente (S/N): ");
32         scanf("%c", &opcao);
33         fflush(stdin); //Função para limpar o buffer do teclado
34         while(opcao!='s' && opcao!='S' && opcao!='n' && opcao!='N')
35         {
36             printf("\nErro! Deseja executar novamente (S/N): ");
37             scanf("%c", &opcao);
38             fflush(stdin); //Função para limpar o buffer do teclado
39         }
40         system("cls");//Limpando a tela
41     }while(opcao=='s' || opcao=='S');
42 }

```

Figura 12.12 – Código do programa em C: respexe125.cpp

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite uma frase: estou aprendendo a programar melhor
A frase "estou aprendendo a programar melhor" possui 5 palavras.
Deseja executar novamente (S/N):

```

Figura 12.13 – Resultado da execução do código do programa em C: respexe125.cpp

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno<a>:

Digite uma frase: Que belo dia
A frase "Que belo dia" possui 3 palavras.
Deseja executar novamente (S/N):
```

Figura 12.14 – Resultado da execução do código do programa em C: *respexe125.cpp*

Fonte: Elaboração própria

12.6 Uma agência espiã deseja enviar mensagens secretas para seus agentes. O código dessas mensagens é a substituição das vogais por números, sendo que as vogais AEIOU correspondem a 23456. Nesse código a frase “EU ESTOU NA AULA DE ICC” corresponde à frase: “36 3ST56 N2 26L2 DE 4CC”. Faça um programa em C que receba uma frase e a codifique-a. A frase deve possuir no máximo 60 caracteres.

Resposta: o código do programa é apresentado na Figura 12.15 e o resultado da execução desse código nas Figuras 12.16 e 12.17.

```

1 #include<string.h> //Biblioteca especifica para funções strings
2 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
3 #include<conio.h> //Biblioteca para manipulação de caracteres
4 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 int main (){//Programa principal
11     // Declaração de variáveis:
12     char frase[100], opcao;
13     int cont;
14     do
15     {
16         //Chamada do procedimento Cabeçalho
17         cabecalho();
18         //Início do Programa
19         printf("Digite uma frase: ");
20         gets(frase);
21         for (cont=0; cont<100; cont++)
22         {
23             if(frase[cont]=='A' || frase[cont]=='a')
24                 frase[cont]='2';
25             else if(frase[cont]=='E' || frase[cont]=='e')
26                 frase[cont]='3';
27             else if(frase[cont]=='I' || frase[cont]=='i')
28                 frase[cont]='4';
29             else if(frase[cont]=='O' || frase[cont]=='o')
30                 frase[cont]='5';
31             else if(frase[cont]=='U' || frase[cont]=='u')
32                 frase[cont]='6';
33         }
34         //Resultado:
35         printf("\nA frase codificada é: %c%c%c.",130,34,frase, 34);
36         printf("\n\nDeseja executar novamente (S/N): ");
37         scanf("%c", &opcao);
38         fflush(stdin); //Função para limpar o buffer do teclado
39         while(opcao!='s' && opcao!='S' && opcao!='n' && opcao!='N')
40         {
41             printf("\n\nErro! Deseja executar novamente (S/N): ");
42             scanf("%c", &opcao);
43             fflush(stdin); //Função para limpar o buffer do teclado
44         }
45         system("cls");//Limpando a tela
46     }while(opcao=='s' || opcao=='S');
47 }

```

Figura 12.15 – Código do programa em C: *respexe126.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite uma frase: EU ESTOU NA AULA DE ICC  
A frase codificada é: "36 3ST56 N2 26L2 D3 4CC".  
Deseja executar novamente <S/N>:
```

Figura 12.16 – Resultado da execução do código programa em C: *respexe126.cpp*

Fonte: Elaboração própria

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite uma frase: APOSTILA DE EXERCICIOS RESOLVIDOS
A frase codificada é: "2P5ST4L2 D3 3X3RC4C45S R3S5LU4D5S".
Deseja executar novamente <S/N>:
```

Figura 12.17 – Resultado da execução do código do programa em C: *respexe126.cpp*

Fonte: Elaboração própria

- 12.7 Escreva um programa em C que se comporte como um vírus. Esse programa irá duplicar as palavras digitadas em uma frase. A frase deve possuir no máximo 30 caracteres.

Resposta: o código do programa é apresentado na Figura 12.18 e o resultado da execução desse código nas Figuras 12.19 e 12.20.

```

1  #include<string.h> //Biblioteca específica para funções strings
2  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
3  #include<conio.h> //Biblioteca para manipulação de caracteres
4  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
5  //Procedimento Cabeçalho
6  void cabecalho(){
7      printf("Universidade de Brasília\n", 161);
8      printf("Disciplina: 113913 - ICC \n");
9      printf("Aluno(a):\n\n");
10 }
11 int main (){//Programa principal
12     // Declaração de variáveis:
13     char frase[30], frasef[60], opcao;
14     int cont, cont2, contff, auxcont;
15     do
16     {
17         //Chamada do procedimento Cabeçalho
18         cabecalho();
19         //Início do Programa
20         printf("Digite uma frase: ");
21         gets(frase);
22         // Estrutura de repetição para andar pela frase
23         for (cont=0, contff=0, auxcont=0; cont< 30; cont++, contff++)
24         {
25             // verificar se é o final da palavra e da frase
26             if((frase[cont]!= ' ') and (frase[cont]!= '\0'))
27                 frasef[contff]=frase[cont];
28                 // se não for o final da frase
29                 // se ainda não for o final da frase final frasef
30                 else if (frasef[contff-1]!='\0')
31                 {
32                     // repete a palavra na frasef
33                     for (cont2 = auxcont ; cont2 < cont; contff++, cont2++)
34                         frasef[contff]=frase[cont2];
35                         // frasef pega o espaço ' ' ou o final da frase '/0'
36                         frasef[contff]=frase[cont];
37                         // auxcont pega o inicio da proxima palavra
38                         auxcont = cont+1;
39                 }
40             }
41             //Resultado:
42             printf("\nVocê digitou %c%c.",34,frasef, 34);
43             printf("\n\nDeseja executar novamente (S/N): ");
44             scanf("%c", &opcao);
45             fflush(stdin); //Função para limpar o buffer do teclado
46             while(opcao!='s' && opcao=='S' && opcao!='n' && opcao=='N')
47             {
48                 printf("\nErro! Deseja executar novamente (S/N): ");
49                 scanf("%c", &opcao);
50                 fflush(stdin); //Função para limpar o buffer do teclado
51             }
52         }
53         system("cls");//Limpando a tela
54     }while(opcao=='s' || opcao=='S');
55 }

```

Figura 12.18 – Código do programa em C: *respexe127.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite uma frase: Hoje é um belo dia
Você digitou "HojeHoje éé umum belobelobelo diadia".
Deseja executar novamente (S/N):
  
```

Figura 12.19 – Resultado da execução do código do programa em C: *respexe127.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite uma frase: Irei no cinema amanhã
Você digitou "IreiIrei nono cinemacinema amanhãamanhã".
Deseja executar novamente (S/N):
  
```

Figura 12.20 – Resultado da execução do código do programa em C: *respexe127.cpp*

Fonte: Elaboração própria

- 12.8 Faça um programa em C que receba uma frase do usuário e que mostre a frase, palavra por palavra, uma em cada linha diferente. A frase deve possuir no máximo 60 caracteres.

Resposta: o código do programa é apresentado na Figura 12.21 e a execução desse código nas Figuras 12.22 e 12.23.

```

1 #include<string.h> //Biblioteca específica para funções strings
2 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
3 #include<conio.h> //Biblioteca para manipulação de caracteres
4 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 int main (){//Programa principal
11     // Declaração de variáveis:
12     char frase[60], opcao;
13     int cont;
14     do
15     {
16         //Chamada do procedimento Cabeçalho
17         cabecalho();
18         //Início do Programa
19         printf("Digite uma frase: ");
20         gets(frase);
21         //Resultado:
22         printf("\nO resultado %c: \n", 130);
23         for (cont=0; cont<strlen(frase); cont++)
24         {
25             if(frase[cont]==' ')
26                 printf("\n");
27             else
28                 printf("%c", frase[cont]);
29         }
30         printf("\n\nDeseja executar novamente (S/N): ");
31         scanf("%c", &opcao);
32         fflush(stdin); //Função para limpar o buffer do teclado
33         while(opcao!='s' && opcao!='S' && opcao!='n' && opcao!='N')
34         {
35             printf("\nErro! Deseja executar novamente (S/N): ");
36             scanf("%c", &opcao);
37             fflush(stdin); //Função para limpar o buffer do teclado
38         }
39         system("cls");//Limpando a tela
40     }while(opcao=='s' || opcao=='S');
41 }

```

Figura 12.21 – Código do programa em C: *respexe128.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite uma frase: Eu estou estudando na Universidade de Brasília
O resultado é:
Eu
estou
estudando
na
Universidade
de
Brasília

Deseja executar novamente (S/N):
  
```

Figura 12.22 – Resultado da execução do código programa em C: *respexe128.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a): 

Digite uma frase: O Brasil é o país do futuro
O resultado é:
O
Brasil
é
o
país
do
futuro

Deseja executar novamente (S/N):
  
```

Figura 12.23 – Resultado da execução do código do programa em C: *respexe128.cpp*

Fonte: Elaboração própria

- 12.9 De aorcdo com uma pqsieusa de uma uinrvesridiae ignlseia, não ipomtra em qaul odrem as lrteas de uma plravaa etāso, a úncia csioa iprotmatne é que a piremria e útmlia lrteas etejasm no lgaur crteo. O rseto pdoe ser uma ttaol bçguana que vcoê cnocseguee anida ler sem pobrlmea. Itso é poqrue nós não lmeos cdaa ltrea szoinha, mas a plravaa

cmoo um tdo. Faça um programa em C que receba uma frase e troque as ordens das letras das palavras conforme o exemplo dessa pesquisa da universidade inglesa. Considere que:

- na frase não haverá pontuação;
- a frase deverá possuir no máximo 60 caracteres;
- as palavras de tamanho iguais podem ser alteradas da mesma forma;
- as palavras terão de tamanho máximo de 8.

Resposta: o código do programa é apresentado na Figura 12.24 e o resultado da execução desse código nas Figuras 12.25 e 12.26.

```

1 #include<string.h>/>Biblioteca especifica para funções strings
2 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
3 #include<conio.h> //Biblioteca para manipulação de caracteres
4 #include<stdlib.h>/>Biblioteca padrão para uso de funções do sistema
5 void cabecalho()
6 {
7     printf("Universidade de Brasília\n", 161);
8     printf("Disciplina: 113913 - ICC \n");
9     printf("Aluno(a):\n\n");
10 }
11 int main ()//Programa principal
12 // Declaração de variáveis:
13 char frase[60], frasef[60], opcao;
14 int cont, contf, auxcont;
15 do
16 {
17     //Chamada do procedimento Cabeçalho
18     cabecalho();
19     //Início do Programa
20     printf("Digite uma frase: ");
21     gets(frase);
22     for (cont=0, contf=0, auxcont=0; cont<60; cont++, contf++)
23     {
24         // verificar se é o final da palavra e da frase
25         if((frase[cont]!= ' ') && (frase[cont]!= '\0'))
26             frasef[contf]=frase[cont];
27             // se não for o final da frase
28             // se ainda não for o final da frase final frasef
29             else if (frasef[contf-1]!= '\0')
30             {
31                 /* A troca da ordem das letras a seguir segui a seguinte regra:
32                 1. Para palavras com o tamanho de um número par, as letras extremas são mantidas. As letras no meio se alternam, das mais extremas
33                 até o meio.
34                 2. Para palavras com o tamanho de um número ímpar, as letras extremas
35                 são mantidas. A letra do meio se mantém também. As alterações

```

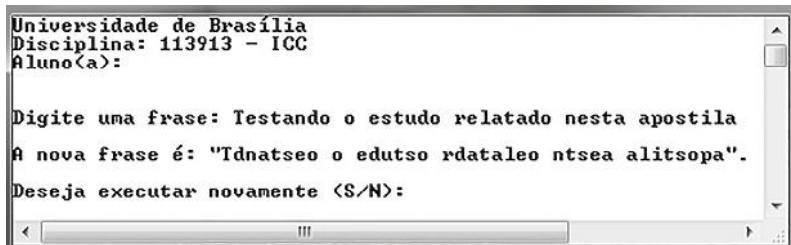
```

36     são realizadas nas letras que não são extremas da palavra, em torno
37     da letra central, onde a alternação ocorre das mais externas até o
38     meio, exceto é claro, com a letra central.
39     Outras regras à gosto poderiam ser definidas.*/
40     if(cont-auxcont==4)
41     {
42         frasef[auxcont+1]=frase[auxcont+2];
43         frasef[auxcont+2]=frase[auxcont+1];
44     }
45     else if(cont-auxcont==5)
46     {
47         frasef[auxcont+1]=frase[auxcont+3];
48         frasef[auxcont+3]=frase[auxcont+1];
49     }
50
51     else if(cont-auxcont==6)
52     {
53         frasef[auxcont+1]=frase[auxcont+4];
54         frasef[auxcont+2]=frase[auxcont+1];
55         frasef[auxcont+3]=frase[auxcont+3];
56         frasef[auxcont+4]=frase[auxcont+2];
57     }
58
59     else if(cont-auxcont==7)
60     {
61         frasef[auxcont+1]=frase[auxcont+5];
62         frasef[auxcont+2]=frase[auxcont+1];
63         frasef[auxcont+3]=frase[auxcont+4];
64         frasef[auxcont+4]=frase[auxcont+2];
65     }
66     else if(cont-auxcont==8)
67     {
68         frasef[auxcont+1]=frase[auxcont+6];
69         frasef[auxcont+2]=frase[auxcont+1];
70         frasef[auxcont+3]=frase[auxcont+5];
71         frasef[auxcont+4]=frase[auxcont+2];
72         frasef[auxcont+5]=frase[auxcont+4];
73         frasef[auxcont+6]=frase[auxcont+3];
74     }
75     frasef[contf]=frase[cont]; //Recebendo o espaço
76     auxcont = cont+1;
77 }
78 }
79 //Resultados:
80 printf("\nA nova frase %c: %c%c%c.",130, 34,frasef, 34);
81 printf("\n\nDeseja executar novamente (S/N): ");
82 scanf("%c", &opcao);
83 fflush(stdin); //Função para limpar o buffer do teclado
84 while(opcao!='s' && opcao!='S' && opcao!='n' && opcao!='N')
85 {
86     printf("\nErro! Deseja executar novamente (S/N): ");
87     scanf("%c", &opcao);
88     fflush(stdin); //Função para limpar o buffer do teclado
89 }
90 system("cls");//Limpando a tela
91 }while(opcao=='s' || opcao=='S');
92 }

```

Figura 12.24 – Código do programa em C: *respexe129.cpp*

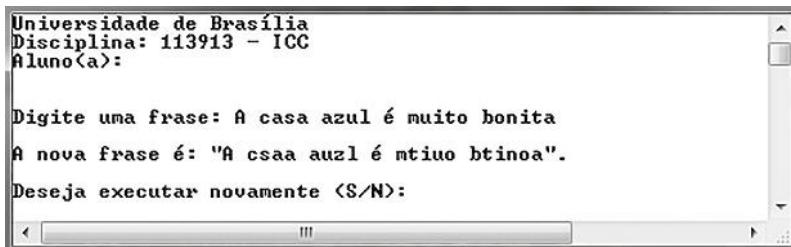
Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite uma frase: Testando o estudo relatado nesta apostila  
A nova frase é: "Tdnatseo o edutso rdataleo ntsea alitsopa".  
Deseja executar novamente (S/N):
```

Figura 12.25 – Resultado da execução do código do programa em C: *respexe129.cpp*

Fonte: Elaboração própria



```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite uma frase: A casa azul é muito bonita
A nova frase é: "A csaa auzl é mtiuo btinoa".
Deseja executar novamente (S/N):
```

Figura 12.26 – Resultado da execução do código do programa em C: *respexe129.cpp*

Fonte: Elaboração própria

Capítulo 13

Ponteiros

Ponteiros

- 13.1** Desenvolva um programa em C que declare duas variáveis do tipo inteiro e duas do tipo ponteiro de inteiro apontando para essas variáveis. Utilizando ponteiros, o programa deve ler dois números para essas variáveis e os imprimir, realizando as quatro operações básicas de matemática com esses números.

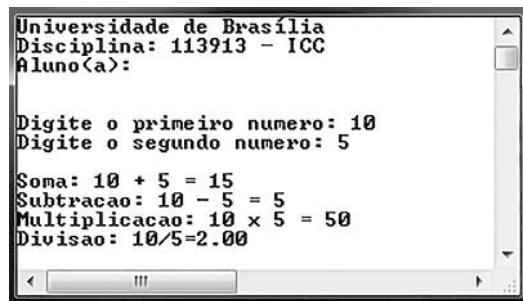
Resposta: o código do programa é apresentado na Figura 13.1 e o resultado da execução desse código na Figura 13.2.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113813 - ICC \n");
8     printf("Aluno(a):\n\n");
9 }
10 int main (){//Programa principal
11     // Declaração de variáveis:
12     int num1,num2;
13     int *pnt1, *pnt2;
14     //Chamada do procedimento Cabeçalho
15     cabecalho();
16     //Início do Programa
17     pnt1=&num1; //Atribuindo endereço da variável ao ponteiro
18     pnt2=&num2; //Atribuindo endereço da variável ao ponteiro
19     printf("Digite o primeiro numero: ");
20     scanf("%d", pnt1); //Recebimento de dados através de ponteiro
21     printf("Digite o segundo numero: ");
22     scanf("%d", pnt2); //Recebimento de dados através de ponteiro
23     //Resultados:
24     /* Trabalhar com a notação de ponteiro: *ponteiro, é trabalhar com
25     o conteúdo que está dentro do endereço que este ponteiro armazena.*/
26     printf("\nSoma: %d + %d = %d", (*pnt1), (*pnt2), (*pnt1)+(*pnt2));
27     printf("\nSubtracao: %d - %d = %d", (*pnt1), (*pnt2), (*pnt1)-(*pnt2));
28     printf("\nMultiplicacao: %d x %d = %d", (*pnt1), (*pnt2), (*pnt1)*(*pnt2));
29     printf("\nDivisao: %d/%d=%f", (*pnt1), (*pnt2), (float)(*pnt1)/(*pnt2));
30
31     getch(); // Função que aguarda o recebimento de um caractere via teclado
32 }
```

Figura 13.1 – Código do programa em C: *respexe131.cpp*

Fonte: Elaboração própria

Figura 13.2 – Resultado da execução do código do programa em C: *respexe131.cpp*

Fonte: Elaboração própria

13.2 Desenvolva um programa em C que declare três variáveis do tipo inteiro e três do tipo ponteiro de inteiro apontando para essas variáveis. Utilizando ponteiros, leia três números e os imprima em ordem crescente. O programa deve apresentar também o endereço de memória desses números.

Resposta: o código do programa é apresentado na Figura 13.3 e o resultado da execução desse código na Figura 13.4.

```

1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h>//Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabecalho
5 void cabecalho(){
6     printf("Universidade de Brasília\n", 161);
7     printf("Disciplina: 113913 - ICC \n");
8     printf("Aluno(a):\n\n\n");
9 }
10 int main ()//Programa principal
11 // Declaração de variáveis:
12 char opcao;
13 int num1,num2, num3;
14 int *pnt1, *pnt2, *pnt3;
15 do
16 {
17     //Chamada do procedimento Cabeçalho
18     cabecalho();
19     //Início do Programa
20     pnt1=&num1; //Atribuindo endereço da variável ao ponteiro

```

```

21     pnt2=&num2; //Atribuindo endereço da variável ao ponteiro
22     pnt3=&num3;
23     printf("Digite o primeiro numero: ");
24     scanf("%d", pnt1); //Recebimento de dados através de ponteiro
25     printf("Digite o segundo numero: ");
26     scanf("%d", pnt2); //Recebimento de dados através de ponteiro
27     printf("Digite o terceiro numero: ");
28     scanf("%d", pnt3); //Recebimento de dados através de ponteiro
29     //Resultados: Obtidos através de comparações dos valores dos ponteiros.
30     /* Trabalhar com a notação de ponteiro: '*ponteiro', é trabalhar com
31     o conteúdo que está dentro do endereço que este ponteiro armazena.*/
32     if(*pnt1>=*pnt2 && *pnt1>=*pnt3)
33     {
34         if(*pnt2>=*pnt3)
35         {
36             printf("\nA ordem crescente dos valores e: %d, %d, %d.", (*pnt3),(*pnt2),(*pnt1));
37             printf("\nOs seus endereços são respectivamente: %p, %p, %p.", pnt3,pnt2,pnt1);
38         }
39         else if(*pnt3>*pnt2)
40         {
41             printf("\nA ordem crescente dos valores e: %d, %d, %d.", (*pnt2),(*pnt3),(*pnt1));
42             printf("\nOs seus endereços são respectivamente: %p, %p, %p.", pnt2,pnt3,pnt1);
43         }
44     }
45     else if(*pnt2>=*pnt1 && *pnt2>=*pnt3)
46     {
47         if(*pnt1>=*pnt3)
48         {
49             printf("\nA ordem crescente dos valores e: %d, %d, %d.", (*pnt3),(*pnt1),(*pnt2));
50             printf("\nOs seus endereços são respectivamente: %p, %p, %p.", pnt3,pnt1,pnt2);
51         }
52         else if(*pnt3>=*pnt1)
53         {
54             printf("\nA ordem crescente dos valores e: %d, %d, %d.", (*pnt1),(*pnt3),(*pnt2));
55             printf("\nOs seus endereços são respectivamente: %p, %p, %p.", pnt1,pnt3,pnt2);
56         }
57     }
58     else if(*pnt3>=*pnt1 && *pnt3>=*pnt2)
59     {
60         if(*pnt1>=*pnt2)
61         {
62             printf("\nA ordem crescente dos valores e: %d, %d, %d.", (*pnt2),(*pnt1),(*pnt3));
63             printf("\nOs seus endereços são respectivamente: %p, %p, %p.", pnt2,pnt1,pnt3);
64         }
65         else if(*pnt2>*pnt3)
66         {
67             printf("\nA ordem crescente dos valores e: %d, %d, %d.", (*pnt1),(*pnt2),(*pnt3));
68             printf("\nOs seus endereços são respectivamente: %p, %p, %p.", pnt1,pnt2,pnt3);
69         }
70     }
71     fflush(stdin); //Função para limpar o buffer do teclado
72     printf("\n\nDeseja executar novamente (S/N): ");
73     scanf("%c", &opcao);
74     while(opcao!='s' && opcao!='S' && opcao!='n' && opcao!='N')
75     {
76         fflush(stdin); //Função para limpar o buffer do teclado
77         printf("\n\nErro! Deseja executar novamente (S/N): ");
78         scanf("%c", &opcao);
79     }
80     system("cls");//Limpa a tela
81 }while(opcao=='s' || opcao=='S');
82

```

Figura 13.3 – Código do programa em C: *respexe132.cpp*

Fonte: Elaboração própria

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite o primeiro numero: 4
Digite o segundo numero: 7
Digite o terceiro numero: 5

A ordem crescente dos valores e: 4, 5, 7.
Os seus enderecos sao respectivamente: 0028FF20, 0028FF18, 0028FF1C.

Deseja executar novamente <S/N>:
!!!

Figura 13.4 – Resultado da execução do código do programa em C: *respexe132.cpp*

Fonte: Elaboração própria

13.3 Desenvolva um programa em C que leia seis números e armazene-os em um vetor. Esse programa deve conter ponteiros para manusear o vetor e imprimir os seus valores. O programa deve apresentar também o endereço de memória desses números.

Resposta: o código do programa é apresentado na Figura 13.5 e o resultado da execução desse código nas Figuras 13.6 e 13.7.

```
1 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2 #include<conio.h> //Biblioteca para manipulação de caracteres
3 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4 //Procedimento Cabeçalho
5 void cabecalho()
6 {
7     printf("Universidade de Brasília\n", 161);
8     printf("Disciplina: 113913 - ICC \n");
9     printf("Aluno(a):\n\n\n");
10 }
11 int main ()//Programa principal
12 // Declaração de variáveis:
13 char opcao;
14 int vet[6], cont;
15 int *pnt;
16 do
17 {
18     //Chamada do procedimento Cabeçalho
19     cabecalho();
20     //Início do Programa
21     pnt=&vet[0]; //Atribuindo endereço do primeiro elemento de vet ao ponteiro
22     printf("Digite 6 números inteiros:\n");
23     for(cont=0; cont<6; cont++)
24     {
25         printf("[%d]: ", cont+1);
26         scanf("%d", pnt); //Recebimento de dados através de ponteiro
27         pnt++; //Indo para a próxima posição da memória
28     }
29     /* Trabalhar com a notação de ponteiro: *ponteiro, é trabalhar com
30     o conteúdo que está dentro do endereço que este ponteiro armazena.
31     *p imprime o endereço apontado*/
32     pnt=&vet[0];//Reatribuindo valor do ponteiro.
33     printf("\nElemento    Valor    Endereço\n\n");
34     for(cont=0; cont<6; cont++)
35     {
```

Figura 13.5 – Código do programa em C: *respexe133.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite 6 números inteiros:
[1]: 1
[2]: 2
[3]: 3
[4]: 4
[5]: 5
[6]: 6

Elemento      Valor      Endereco
1              1          0028FEF0
2              2          0028FEF4
3              3          0028FEF8
4              4          0028FEFC
5              5          0028FF00
6              6          0028FF04

Deseja executar novamente <S/N>:

```

Figura 13.6 – Resultado da execução do código do programa em C: respexe133.cpp

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite 6 números inteiros:
[1]: 11
[2]: 45
[3]: 64
[4]: 32
[5]: 78
[6]: 90

Elemento      Valor      Endereco
1              11         0028FEF0
2              45         0028FEF4
3              64         0028FEF8
4              32         0028FEFC
5              78         0028FF00
6              90         0028FF04

Deseja executar novamente <S/N>:

```

Figura 13.7 – Resultado da execução do código do programa em C: respexe133.cpp

Fonte: Elaboração própria

- 13.4 Desenvolva um programa em C que leia quatro números e armazene-os em um vetor. Esse programa deve conter ponteiros para manusear o vetor e imprimir os seus valores. Esse programa deve conter ponteiros também para apresentar o maior e o menor número do vetor.

Resposta: o código do programa é apresentado na Figura 13.8 e o resultado da execução desse código na Figura 13.9.

```

1  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
2  #include<conio.h> //Biblioteca para manipulação de caracteres
3  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
4  //Procedimento Cabeçalho
5  void cabecalho(){
6      printf("Universidade de Brasília\n", 161);
7      printf("Disciplina: 113913 - ICC \n");
8      printf("Aluno(a):\n\n");
9  }
10 int main (){//Programa principal
11     // Declaração de variáveis:
12     char opcao;
13     int vet[], cont, maior, menor;
14     int *pnt, *pntMaior, *pntMenor;
15     do
16     {
17         //Chamada do procedimento Cabeçalho
18         cabecalho();
19         //Início do Programa
20         //Atribuindo endereço do primeiro elemento de vet ao ponteiro
21         pnt=&vet[0];
22         pntMaior=&maior;
23         pntMenor=&menor;
24         //Atribuindo os valores iniciais guardados nos endereços apontados
25         maior=-9999;
26         menor=9999;
27         printf("Digite 4 valores:\n");
28         for(cont=0; cont<4; cont++)
29         {
30             printf("[%d]: ", cont+1);
31             scanf("%d", pnt); //Recebimento de dados através de ponteiro
32             if(*pnt>*pntMaior)
33                 pntMaior=pnt; //Atribuindo o mesmo endereço
34             if(*pnt<*pntMenor)
35                 pntMenor=pnt; //Atribuindo o mesmo endereço
36             pnt++; //Indo para a próxima posição da memória
37         }
38         //Resultados:
39         printf("\nO maior elemento do vetor é: %d", *pntMaior);
40         printf("\nO menor elemento do vetor é: %d", *pntMenor);
41
42         fflush(stdin); //Função para limpar o buffer do teclado
43         printf("\n\nDeseja executar novamente (S/N): ");
44         scanf("%c", &opcao);
45         while(opcao!='s' && opcao!='N' && opcao!='n')
46         {
47             fflush(stdin); //Função para limpar o buffer do teclado
48             printf("\n\nErro! Deseja executar novamente (S/N): ");
49             scanf("%c", &opcao);
50         }
51         system("cls");//Limpando a tela
52     }while(opcao=='s' || opcao=='S');
53 }
```

Figura 13.8 – Código do programa em C: *respexe134.cpp*

Fonte: Elaboração própria

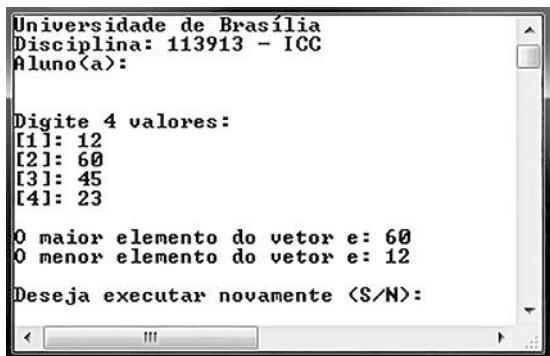


Figura 13.9 – Resultado da execução do código do programa em C: respexe134.cpp

Fonte: Elaboração própria

13.5 Faça um programa em C que receba uma frase qualquer fornecida pelo usuário, que calcule e mostre quantos caracteres essa frase possui. Esse programa deve conter ponteiros para manusear a string. A frase deve possuir no máximo 50 caracteres.

Resposta: o código do programa é apresentado na Figura 13.10 e o resultado da execução desse código na Figura 13.11.

```

1 #include<string.h>/Biblioteca especifica para funções strings
2 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
3 #include<conio.h> //Biblioteca para manipulação de caracteres
4 #include<stdlib.h>/Biblioteca padrão para uso de funções do sistema
5 //Procedimento Cabecalho
6 void cabecalho(){
7     printf("Universidade de Brasília\n", 161);
8     printf("Disciplina: 113913 - ICC \n");
9     printf("Aluno(a):\n\n");
10 }
11 int main (){//Programa principal
12     // Declaração de variáveis:
13     char opcao, frase[50];
14     int cont, qnt, tamanho;
15     char *pnt;
16     do
17     {
18         //Chamada do procedimento Cabeçalho
19         cabecalho();
20         //Início do Programa
21         pnt=&frase[0]; //Atribuindo endereço do primeiro elemento de vet ao ponteiro
22         qnt=0;
    
```

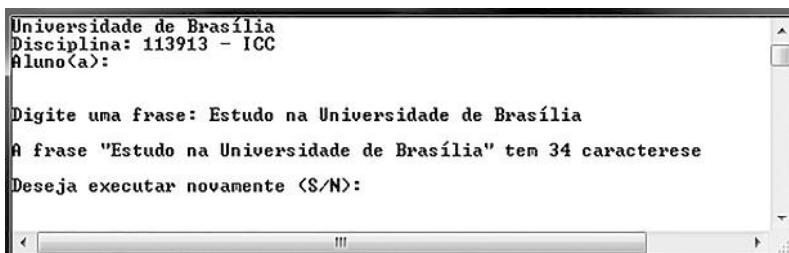
```

23 //Recebendo frase por ponteiros
24 /* Para caracteres, a forma de manipulação de ponteiros é a mesma. A única
25 exigência para correta utilização de ponteiros para caracteres é indicar na
26 sua declaração que este armazenará um valor do tipo 'char'.*/
27 printf("Digite uma frase: ");
28 gets(frase);
29 tamanho = strlen(frase);
30 while((frase[0] == '\0') || tamanho > 100) // validação para tamanho de frase
31 {
32     puts("\nErro! Digite uma frase: ");
33     gets(frase); //Função que recebe uma string
34     tamanho = strlen(frase);
35 }
36 for(cont=0; cont<50; cont++)
37 {
38     if(*pnt=='\0') //Comparando a posição atual do vetor
39         break; //Finaliza o laço
40     qnt++;
41     pnt++; //Indo para a próxima posição da memória
42 }
43
44 //Resultados:
45 pnt=&frase[0];
46 printf("\nA frase %c", 34);
47 for(cont=0: frase[cont]!='\0': cont++)
48 {
49     printf("%c", *pnt);
50     pnt++;
51 }
52 printf("%c tem %d caracterese", 34, qnt);
53 printf("\n\nDeseja executar novamente (S/N): ");
54 scanf("%c", &opcao);
55 fflush(stdin); //Função para limpar o buffer do teclado
56 while(opcao!='s' && opcao!='S' && opcao!='n' && opcao!='N')
57 {
58     printf("\n\nErro! Deseja executar novamente (S/N): ");
59     scanf("%c", &opcao);
60     fflush(stdin); //Função para limpar o buffer do teclado
61 }
62 system("cls");//Limpando a tela
63 }while(opcao=='s' || opcao=='S');
64 }

```

Figura 13.10 – Código do programa em C: *respexe135.cpp*

Fonte: Elaboração própria

Figura 13.11 – Resultado da execução do código do programa em C: *respexe135.cpp*

Fonte: Elaboração própria

- 13.6 Faça um programa em C que receba uma frase qualquer fornecida pelo usuário, calcule e mostre quantas palavras essa frase possui. Esse programa deve conter ponteiros para manusear a string. A frase deve possuir no máximo 50 caracteres.

Resposta: o código do programa é apresentado na Figura 13.12 e a execução desse código na Figura 13.13.

```

1  #include<string.h> //Biblioteca específica para funções strings
2  #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
3  #include<conio.h> //Biblioteca para manipulação de caracteres
4  #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
5  //Procedimento Cabecalho
6  void cabecalho()
7  {
8      printf("Universidade de Brasília\n", 161);
9      printf("Disciplina: 113913 - ICC \n");
10     printf("Aluno(a):\n\n");
11 }
12 int main ()//Programa principal
13 // Declaração de variáveis:
14 char opcao, frase[50];
15 int cont, qnt, tamanho;
16 char *pnt;
17 do
18 {
19     //Chamada do procedimento Cabecalho
20     cabecalho();
21     //Início do Programa
22     pnt=&frase[0]; //Atribuindo endereço do primeiro elemento de vet ao ponteiro
23     qnt=1; //Conta os espaços. Quando surge o primeiro espaço já se tem uma palavra
24     //Recebendo frase por ponteiros
25     /* Para caracteres, a forma de manipulação de ponteiros é a mesma. A única
26     exigência para correta utilização de ponteiros para caracteres é indicar na
27     sua declaração que este armazenará um valor do tipo 'char'.*/
28     printf("Digite uma frase: ");
29     gets(frase);
30     tamanho = strlen(frase);
31     while((frase[0] == '\0') || tamanho > 100) // validação para tamanho de frase
32     {
33         puts("\nErro! Digite outra frase: ");
34         gets(frase); //Função que recebe uma string
35         tamanho = strlen(frase);
36     }
37     for(cont=0; cont<50; cont++)
38     {
39         if(*pnt==' ') //Comparando a posição atual do vetor
40             qnt++;
41         pnt++; //Indo para a próxima posição da memória
42     }
43
44     //Resultados:
45     pnt=&frase[0];
46     printf("\nA frase %c", 34);
47     for(cont=0; frase[cont]!='\0'; cont++)

```

```

47     {
48         printf("%c", *pnt);
49         pnt++;
50     }
51     printf("%c tem %d palavras", 34, qnt);
52     printf("\n\nDeseja executar novamente (S/N): ");
53     scanf("%c", &opcao);
54     fflush(stdin); //Função para limpar o buffer do teclado
55     while(opcao!='s' && opcao!='S' && opcao!='n' && opcao!='N')
56     {
57         printf("\n\nErro! Deseja executar novamente (S/N): ");
58         scanf("%c", &opcao);
59         fflush(stdin); //Função para limpar o buffer do teclado
60     }
61     system("cls");//Limpo a tela
62 }while(opcao=='s' || opcao=='S');
63

```

Figura 13.12 – Código do programa em C: *respexe136.cpp*

Fonte: Elaboração própria

```

Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):

Digite uma frase: Estudando para aprender a programar bem
A frase "Estudando para aprender a programar bem" tem 6 palavras
Deseja executar novamente <S/N>:

```

Figura 13.13 – Resultado da execução do código do programa em C: *respexe136.cpp*

Fonte: Elaboração própria

- 13.7 Faça um programa em C que receba uma frase do usuário e mostre a frase, palavra por palavra, uma em cada linha diferente. Esse programa deve conter ponteiros na para manusear a string. A frase deve possuir no máximo 60 caracteres.

Resposta: o código do programa é apresentado na Figura 13.14 e o resultado da execução desse código na Figura 13.15.

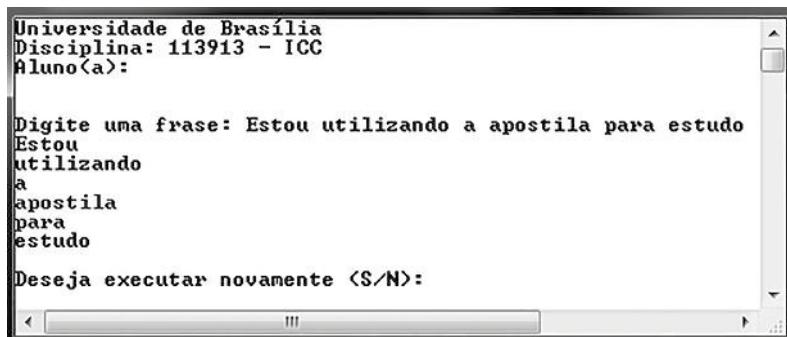
```

1 #include<string.h> //Biblioteca específica para funções strings
2 #include<stdio.h> //Biblioteca de entrada/saída por periféricos padrão
3 #include<conio.h> //Biblioteca para manipulação de caracteres
4 #include<stdlib.h> //Biblioteca padrão para uso de funções do sistema
5 void cabecalho()
6 {
7     printf("Universidade de Brasília\n", 161);
8     printf("Disciplina: 113913 - ICC \n");
9     printf("Aluno(a):\n\n\n");
10 }
11 int main ()//Programa principal
12 // Declaração de variáveis:
13 char frase[60], opcao, tamanho;
14 char *pnt;
15 int cont;
16 do
17 {
18     //Chamada do procedimento Cabeçalho
19     cabecalho();
20     //Início do Programa
21     pnt=&frase[0]; //Atribuindo endereço do primeiro elemento de vet ao ponteiro
22     printf("Digite uma frase: ");
23     gets(frase);
24     tamanho = strlen(frase);
25     while((frase[0] == '\0') || tamanho > 100) // Validação para tamanho de frase
26     {
27         puts("\nErro! Digite uma frase menor: ");
28         gets(frase); //Função que recebe uma string
29         tamanho = strlen(frase);
30     }
31     //Resultados:
32     for (cont=0; cont<strlen(frase); cont++)
33     {
34         if(*pnt==' ')
35             printf("\n");
36         else
37             printf("%c", *pnt);
38         pnt++;
39     }
40     printf("\n\nDeseja executar novamente (S/N): ");
41     scanf("%c", &opcao);
42     fflush(stdin); //Função para limpar o buffer do teclado
43     while(opcao!='s' && opcao!='S' && opcao!='n' && opcao!='N')
44     {
45         printf("\nErro! Deseja executar novamente (S/N): ");
46         scanf("%c", &opcao);
47         fflush(stdin); //Função para limpar o buffer do teclado
48     }
49     system("cls");//Limpando a tela
50 }while(opcao=='s' || opcao=='S');
51 }

```

Figura 13.14 – Código do programa em C: *respexe137.cpp*

Fonte: Elaboração própria



The screenshot shows a terminal window with the following text output:

```
Universidade de Brasília
Disciplina: 113913 - ICC
Aluno(a):  
  
Digite uma frase: Estou utilizando a apostila para estudo
Estou
utilizando
a
apostila
para
estudo  
  
Deseja executar novamente (S/N):
```

Figura 13.15 – Resultado da execução do código do programa em C: *respexe137.cpp*

Fonte: Elaboração própria

Bibliografia

BANAHAN, Mike; BRADY, Declan; DORAN, Mark. **The C book.** 2. ed. Boston: Addison-Wesley, 2003.

DAMAS, Luis M. Dias. **Linguagem C.** 10. ed. Rio de Janeiro: LTC, 2007.

DEITEL, Paul; DEITEL, Harvey M. **C: how to program.** 7. ed. Nova Iorque: Pearson Education, 2013.

EUER, Alan R. **The C Puzzle Book.** 1. ed. Boston: Addison-Wesley, 1998.

GRIFFITHS, David; GRIFFITHS, Dawn. **Head First C.** 6. ed. Boston: O'Reilly Media, 2012.

HANSON, David R. **C Interfaces and Implementations.** 3. ed. Boston: Addison-Wesley, 2001.

KLEMENS, Ben. **21st Century C.** 2. ed. Boston: O'Reilly Media, 2014.

KOCHAN, Stephen G. **Programming in C.** 3. ed. Indianapolis: Sam's Publishing, 2005.

LINDEN, Peter van der. **Expert C Programming: deep c secrets.** 1 ed. Nova Iorque: Prentice Hall, 1994.

- MIZRAHI, Victorine Viviane. **Treinamento em linguagem C.** 2. ed. São Paulo: Pearson Prentice Hall, 2008.
- OUALLINE, Steve. **Practical C Programming.** 3. ed. Boston: O'Reilly Media, 1997.
- PERRY, John W. **Advanced C Programming by Example.** 1. ed. [S.l.]: Pws Pub Co, 1998.
- PLAUGER, P. J. **The Standard C Library.** 1. ed. Nova Iorque: Prentice Hall, 1992.
- PRATA, Stephen. **C Primer Plus.** 6. ed. Boston: Addison-Wesley, 2014.
- REEK, Kenneth. **Pointers on C.** 1. ed. Nova Iorque: Pearson, 1997.
- SCHILDIT, Herbert. **C completo e total.** 3. ed. São Paulo: Makron Books, 1997.
- SCHREINER, Axel-Tobias. **Object-oriented programming with ANSI-C.** 1999. Disponível em: <<https://www.cs.rit.edu/~ats/books/ooc.pdf>>. Acesso em: 5 set. 2016.
- SEDGEWICK, Robert. **Algorithms in C.** 3. ed. Boston: Addison-Wesley, 1997.

Esta obra objetiva auxiliar estudantes ou interessados em aprender a programar em linguagem C por meio de exercícios resolvidos de programação. Este livro pode ser usado nas aulas de laboratório em linguagem de cursos de graduação nas áreas de exatas e por pessoas que desejam aprender e praticar os conceitos da linguagem C. Neste livro, os conteúdos de programação em linguagem C estão organizados em 13 capítulos, na mesma ordem do volume 1 dessa Coleção.

Este volume apresenta um exemplo simples de solução para os exercícios das atividades de fixação do volume 1 dessa Coleção, e apresenta também outros exercícios e suas soluções. Este livro contém mais de 100 exercícios resolvidos que estão ordenados com diferentes níveis de dificuldade, possibilitando ao estudante um desenvolvimento gradual e consistente.

Este livro completa a “Coleção Introdução à Linguagem C”, cujo volume anterior é o livro “Roteiro de Experimentos para Aulas Práticas”.

ISBN: 978-85-79883-10-1
(Coleção Completa)



ISBN: 978-85-7988-329-3

