

Acadêmica: Ana Paula Antunes Araujo
Turma: Hands on work I
1º semestre de 2021

○ HANDS ON WORK ○

| UNIVALI |

○ Aplicação console em C que realize conversões de base ○
numéricas

CONCEITOS APRENDIDOS

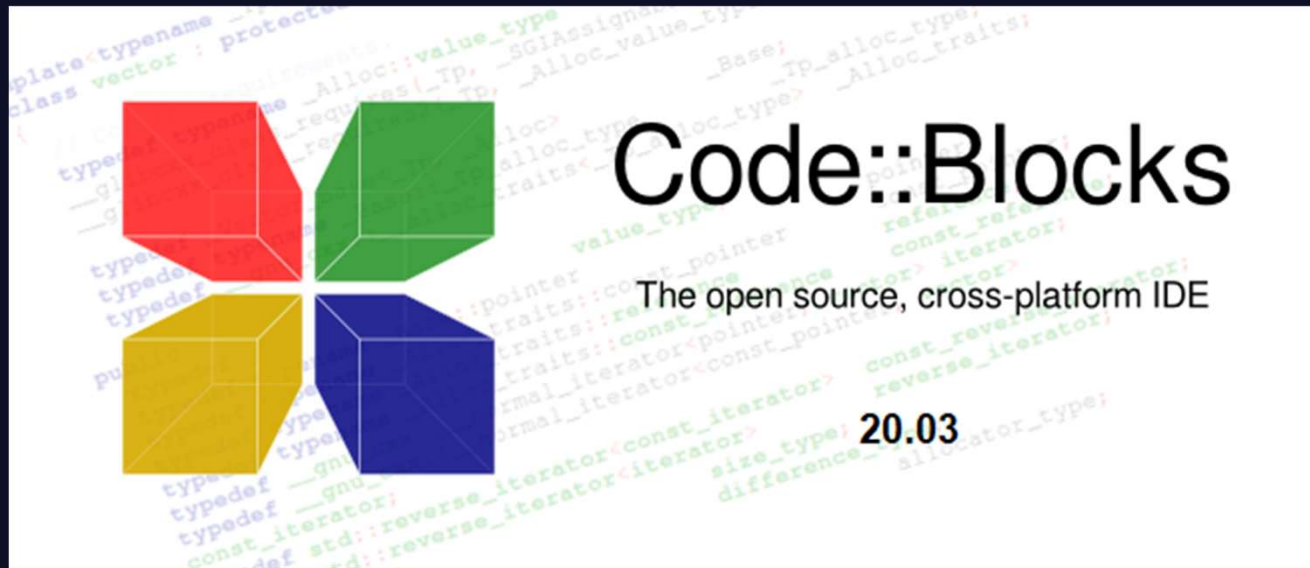
- **Lógica de programação:** Capacidade de resolver problemas, dividindo em partes menores;
- **Sistemas de numeração:** Conjuntos de símbolos e regras para representar o número de elementos de conjuntos quaisquer;
- **Conversão de base numérica:** Cálculos necessário para realizar a conversão entre as bases numéricas;
- **Linguagem C:** Linguagem criada por volta de 1970 que se caracteriza pela sua portabilidade, poder e variedade dos operadores, sintaxe elegante, estruturada e flexível, acesso facilitado a memória e a todo hardware, uso de procedimentos e funções para desenvolver sistemas desacoplados.

PROJETO

Vamos agora ao projeto elaborado utilizando esses conceitos.

CODEBLOCKS

O código fonte foi escrito utilizando o CodeBlocks.
Todo o projeto foi elaborado apenas com esta ferramenta.



CODIGO FONTE

BIBLIOTECAS

`#include <stdio.h>` = Cabeçalho da biblioteca padrão do C. Seu nome vem da expressão inglesa standard input-output header, que significa "cabeçalho padrão de entrada/saída". Permite a utilização das funções `printf`, `scanf`, `fprintf`, `fscanf`.

`#include <math.h>` = Conjunto de funções para operações matemáticas, tais como funções trigonométricas, hiperbólicas, logaritmos, potência e arredondamentos.

`#include <string.h>` = Funções para manipular strings.

```
#include <stdio.h>
#include <math.h>
#include <string.h>
```

CODIGO FONTE

VARIÁVEIS

As variáveis declaradas foram as seguintes:

- Opcao = guarda a opção selecionada pelo usuário;
- Valor_octal = guarda o valor octal da conversão que estiver sendo realizada;
- Opcao 2 = Utilizada no laço de repetição que permite realizar nova conversão ou fechar a aplicação;
- Valor_binário = guarda o valor binário da conversão que estiver sendo realizada;
- Valor decimal = guarda o valor decimal da conversão que estiver sendo realizada;
- Sequencial = recebe o valor de sequencial que será utilizado nos cálculos para conversão.

```
//VARIÁVEIS UTILIZADAS
char opcao = ' ';
int valor_octal;
int opcao2 = 1;
long long valor_binario;
int valor_decimal;
int sequencial;
```

CODIGO FONTE

FUNÇÕES

Foram criadas funções para realizar os cálculos.
A chamada dessas funções foi realizada fora da função main();
A seguir mostrarei cada uma delas realizará os cálculos de conversão.

```
//DECLARAÇÃO OU PROTÓTIPO DAS FUNÇÕES UTILIZADAS  
//A DECLARACAO É FEITA DA SEGUINTE FORMA:  
//TIPO + NOME + (LISTA DE PARÂMETROS QUEA FUNÇÃO PRECISA  
int converterBinDec(long long valor_binario);  
long long converterDecBin(int valor_decimal);  
int converterDecOct(int valor_decimal);  
long long converterOctDec(int valor_octal);  
  
int main()  
{
```

BINÁRIO PARA DECIMAL

```
//CONVERSÃO DE BINARIO X DECIMAL
int converterBinDec(long long valor_binario)
{
    //VARIÁVEIS UTILIZADAS NA FUNCAO
    int valor_decimal = 0, sequencial = 0, resto;

    //LAÇO DE REPETIÇÃO
    //ENQUANTO EXISTIR VALOR NO BINÁRIO ZERO
    while (valor_binario != 0)
    {
        //PEGA O RESTO DA DIVISÃO DO VALOR POR 10
        resto = valor_binario % 10;

        //DIVIDE O VALOR BINARIO POR 10
        valor_binario /= 10;

        //INCREMENTA O VALOR DECIMAL COM O RESTO DA DIVISÃO MULTIPLICANDO POR 2 ELEVADO AO SEQUENCIAL
        //VALOR DECIMAL É IGUAL A SOMA DE (OS RESTOS MULTIPLICADOS PELO NUMERO SEQUENCIAL ELEVADO A 2)
        valor_decimal += resto * pow(2, sequencial);

        //INCREMENTA A SEQUENCIAL
        ++sequencial;
    }

    return valor_decimal;
}
```

Para converter de binário para decimal a função que realiza essa conversão executa os seguintes cálculos:

- Consegue o resto da divisão do valor binário por 10 e salva na variável resto;
- Divide o valor binário e salva o resultado na variável valor_binario;
- Pega 2 elevado pelo valor sequencial, o resultado multiplica pelo valor da variável resto, o resultado dessa multiplicação soma com o valor da variável valor_decimal;
- Incrementa 1 na variável sequencial.

Esses cálculos serão realizados repetidas vezes até que a variável valor_binario seja igual a zero.

- Vamos ver na prática como essa função converteria o valor binário 00010001
-

Sequencial = 0

Valor decimal = 0

Resto -> $10001 \% 10 = 1$

Valor_binario -> $10001 / 10 = 1000$

Com esses podemos calcular o decimal da seguinte forma:

$\text{Valor_decimal} = (1 * (2)^0) + 0$

$\text{Valor_decimal} = (1 * 1) + 0$

$\text{Valor_decimal} = 1 + 0$

$\text{Valor_decimal} = 1$

Após este calculo o valor da variável sequencial é incrementado e passa a ser 1.

E com isso temos as variáveis conforme quadro ao lado

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	1
Valor_binario	1000
Sequencial	1
Resto	1

- Vamos ver na prática como essa função converteria o valor binário 00010001
-

Sequencial = 1

Valor decimal = 1

Resto -> $1000,1 \% 10 = 0$

Valor_binario -> $1000 / 10 = 100$

$\text{Valor_decimal} = (0 * (2)^1) + 1$

$\text{Valor_decimal} = (0 * 2) + 1$

$\text{Valor_decimal} = 0 + 1$

$\text{Valor_decimal} = 1$

Por fim é incrementado 1 no valor da variável sequencial

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	1
Valor_binario	100
Sequencial	2
Resto	0

- Vamos ver na prática como essa função converteria o valor binário 00010001
-

Sequencial = 2

Valor decimal = 1

Resto -> $100 \% 10 = 0$

Valor_binario -> $100 / 10 = 10$

Valor_decimal = $(0 * (2)^2) + 1$

Valor_decimal = $(0 * 4) + 1$

Valor_decimal = $0 + 1$

Valor_decimal = 1

Por fim é incrementado 1 no valor da variável sequencial

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	1
Valor_binario	10
Sequencial	3
Resto	0

- Vamos ver na prática como essa função converteria o valor binário 00010001
-

Sequencial = 3

Valor decimal = 1

Resto -> $10 \% 10 = 0$

Valor_binario -> $10 / 10 = 1$

$\text{Valor_decimal} = (0 * (2)^3) + 1$

$\text{Valor_decimal} = (0 * 8) + 1$

$\text{Valor_decimal} = 0 + 1$

$\text{Valor_decimal} = 1$

Por fim é incrementado 1 no valor da variável sequencial

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	1
Valor_binario	1
Sequencial	4
Resto	0

- Vamos ver na prática como essa função converteria o valor binário 00010001
-

Sequencial = 4

Valor decimal = 1

Resto -> $1 \% 10 = 1$

Valor_binario -> $1 / 10 = 0$

$\text{Valor_decimal} = (1 * (2)^4) + 1$

$\text{Valor_decimal} = (1 * 16) + 1$

$\text{Valor_decimal} = 16 + 1$

$\text{Valor_decimal} = 17$

Por fim é incrementado 1 no valor da variável sequencial

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	17
Valor_binario	0
Sequencial	5
Resto	1

DECIMAL PARA BINÁRIO

```
//CONVERSAO DE DECIMAL X BINARIO
long long converterDecBin(int valor_decimal)
{
    //VARIÁVEIS UTILIZADAS NA FUNCAO
    long long valor_binario = 0;
    int resto, sequencial = 1;

    //LAÇO DE REPETIÇÃO
    //ENQUANTO O VALOR DECIMAL FOR DIFERENTE DE ZERO
    while (valor_decimal != 0)
    {
        //PEGA O RESTO DA DIVISAO
        resto = valor_decimal % 2;

        //DIVIDE O VALOR DECIMAL POR 2
        valor_decimal /= 2;

        //INCREMENTA O VALOR BINÁRIO, MULTIPLICANDO O RESTO DA DIVISAO PELO SEQUENCIAL
        valor_binario += resto * sequencial;

        //MULTIPLICANDO O SEQUENCIAL POR 10
        sequencial *= 10;
    }

    return valor_binario;
}
```

Para converter de decimal para binário a função que realiza essa conversão executa os seguintes cálculos:

- Consegue o resto da divisão do valor decimal por 2 e salva na variável resto;
- Divide o valor decimal por 2 e salva o resultado na variável valor_decimal;
- Pega o valor sequencial e multiplica pelo resto e soma com o valor binário anterior;
- Incrementa a variável sequencial multiplicando por 10.

Esses cálculos serão realizados repetidas vezes até que a variável valor_decimal seja igual a zero.

- Vamos ver na prática como essa função converteria o valor decimal 17 ○

Sequencial = 1

Resto -> $17\%2 = 1$

Valor_decimal -> $17/2 = 8$

Com esses podemos calcular o decimal da seguinte forma:

$\text{Valor_binario} = (1 * 1) + 0$

$\text{Valor_binario} = 1 + 0$

$\text{Valor_binario} = 1$

Após este calculo o valor da variável sequencial é incrementado e passa a ser 10.

E com isso temos as variáveis conforme quadro ao lado

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	8
Valor_binario	1
Sequencial	10
Resto	1

- Vamos ver na prática como essa função converteria o valor decimal 17 ○

Sequencial = 10

Resto $\rightarrow 8\%2 = 0$

Valor_decimal $\rightarrow 8/2 = 4$

Com esses podemos calcular o decimal da seguinte forma:

Valor_binario = $(0 * 10) + 1$

Valor_binario = $0 + 1$

Valor_binario = 1

Após este calculo o valor da variável sequencial é incrementado e passa a ser 100.

E com isso temos as variáveis conforme quadro ao lado

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	4
Valor_binario	1
Sequencial	100
Resto	0

- Vamos ver na prática como essa função converteria o valor decimal 17 ○

Sequencial = 100

Resto $\rightarrow 4 \% 2 = 0$

Valor_decimal $\rightarrow 4 / 2 = 2$

Com esses podemos calcular o decimal da seguinte forma:

Valor_binario = $(0 * 100) + 1$

Valor_binario = $0 + 1$

Valor_binario = 1

Após este calculo o valor da variável sequencial é incrementado e passa a ser 1000.

E com isso temos as variáveis conforme quadro ao lado

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	2
Valor_binario	1
Sequencial	1000
Resto	0

- Vamos ver na prática como essa função converteria o valor decimal 17 ○

Sequencial = 1000

Resto $\rightarrow 2\%2 = 0$

Valor_decimal $\rightarrow 2/2 = 1$

Com esses podemos calcular o decimal da seguinte forma:

$\text{Valor_binario} = (0 * 1000) + 1$

$\text{Valor_binario} = 0 + 1$

$\text{Valor_binario} = 1$

Após este calculo o valor da variável sequencial é incrementado e passa a ser 10000.

E com isso temos as variáveis conforme quadro ao lado

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	1
Valor_binario	1
Sequencial	10000
Resto	0

- Vamos ver na prática como essa função converteria o valor decimal 17 ○

Sequencial = 10000

Resto -> $1\%2 = 1$

Valor_decimal -> $1/2 = 0$

Com esses podemos calcular o decimal da seguinte forma:

$\text{Valor_binario} = (1 * 10000) + 1$

$\text{Valor_binario} = 10000 + 1$

$\text{Valor_binario} = 10001$

Após este calculo o valor da variável sequencial é incrementado e passa a ser 100000.

E com isso temos as variáveis conforme quadro ao lado

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	0
Valor_binario	10001
Sequencial	100000
Resto	1

OCTAL PARA DECIMAL

```
//CONVERSAO DE OCTAL X DECIMAL
long long converterOctDec(int valor_octal)
{
    //VARIÁVEIS UTILIZADAS NA FUNCAO
    int valor_decimal = 0, sequencia = 0;

    //LAÇO DE REPETIÇÃO
    //ENQUANTO O VALOR OCTAL FOR DIFERENTE DE ZERO
    while(valor_octal != 0)
    {
        //INCREMENTA O VALOR DECIMAL COM O RESTO DA DIVISÃO DO VALOR
        //OCTAL POR 10 MULTIPLICADO POR 8 ELEVADO PELO SEQUENCIAL
        valor_decimal += (valor_octal % 10) * pow(8, sequencia);

        //INCREMENTA O SEQUENCIAL
        ++sequencia;

        //DIVIDE O VALOR OCTAL POR 10
        valor_octal /= 10;
    }

    return valor_decimal;
}
```

Para converter de octal para decimal a função que realiza essa conversão executa os seguintes cálculos:

- Consegue o resto da divisão do valor octal por 10 e multiplica por 8 elevado ao numero sequencial, o resultado soma com o valor decimal anterior;
- Incrementa 1 na variável sequencia.
- Divide o valor octal por 10 para conseguir o novo valor da variável valor_octal;

Esses cálculos serão realizados repetidas vezes até que a variável valor_octal seja igual a zero.

- Vamos ver na prática como essa função converteria o valor octal 21 ○

Sequencia = 0

Valor_ decimal -> 0

Com esses podemos calcular o decimal da seguinte forma:

$\text{Valor_decimal} = ((21 \bmod 10) * (8)^0) + 0$

$\text{Valor_decimal} = (1 * 1) + 0$

$\text{Valor_decimal} = 1 + 0$

$\text{Valor_decimal} = 1$

A variável sequencia é incrementada, somando 1 ao seu valor.

$\text{Valor_octal} = 21/10 = 2$

E com isso temos as variáveis conforme quadro ao lado

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	1
Valor_octal	2
Sequencia	1

- Vamos ver na prática como essa função converteria o valor octal 21
-

Sequencia = 1

Valor_ decimal -> 1

Com esses podemos calcular o decimal da seguinte forma:

$\text{Valor_decimal} = ((2 \bmod 10) * (8)^1) + 1$

$\text{Valor_decimal} = (2 * 8) + 1$

$\text{Valor_decimal} = 16 + 1$

$\text{Valor_decimal} = 17$

A variável sequencia é incrementada, somando 1 ao seu valor.

$\text{Valor_octal} = 2/10 = 0$

E com isso temos as variáveis conforme quadro ao lado

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	17
Valor_octal	0
Sequencia	2

DECIMAL PARA OCTAL

```
//CONVERSAO DE DECIMAL X OCTAL
int converterDecOct(int valor_decimal)
{
    //VARIÁVEIS UTILIZADAS NA FUNCAO
    int valor_octal = 0, sequencia = 1;

    //LAÇO DE REPETIÇÃO
    //ENQUANTO O VALOR DECIMAL FOR DIFERENTE DE ZERO
    while (valor_decimal != 0)
    {
        //INCREMENTA O VALOR OCTAL COM O RESTO DA DIVISAO
        //DO DECIMAL POR 8 MULTIPLICANDO PELO SEQUENCIAL
        valor_octal += (valor_decimal % 8) * sequencia;

        //O VALOR DECIMAL SERÁ DIVIDIDO POR 8
        valor_decimal /= 8;

        //O SEQUENCIAL SERÁ MULTIPLICADO POR 10
        sequencia *= 10;
    }

    return valor_octal;
}
```

Para converter de decimal para octal a função que realiza essa conversão executa os seguintes cálculos:

- Consegue o resto da divisão do valor decimal por 8 e multiplica pela variável sequencia, o resultado soma com o valor octal anterior;
- Incrementa a variável sequencia multiplicando por 10.
- Divide o valor decimal por 8 para conseguir o novo valor da variável valor_decimal;

Esses cálculos serão realizados repetidas vezes até que a variável valor_decimal seja igual a zero.

- Vamos ver na prática como essa função converteria o valor decimal 17
-

Sequencia = 1

Valor_octal = 0

Com esses podemos calcular o decimal da seguinte forma:

$\text{Valor_octal} = ((17 \bmod 8) * 1) + 0$

$\text{Valor_octal} = (1 * 1) + 0$

$\text{Valor_octal} = 1 + 0$

$\text{Valor_octal} = 1$

A variável sequencia é incrementada, multiplicando por 10 ao seu valor.

$\text{Valor_decimal} = 17/8 = 2$

E com isso temos as variáveis conforme quadro ao lado

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	2
Valor_octal	1
Sequencia	10

- Vamos ver na prática como essa função converteria o valor decimal 17
-

Sequencia = 10

Valor_octal = 1

Com esses podemos calcular o decimal da seguinte forma:

$\text{Valor_octal} = ((2 \bmod 8) * 10) + 1$

$\text{Valor_octal} = (2 * 10) + 1$

$\text{Valor_octal} = 20 + 1$

$\text{Valor_octal} = 21$

A variável sequencia é incrementada, multiplicando por 10 ao seu valor.

$\text{Valor_decimal} = 2/8 = 0$

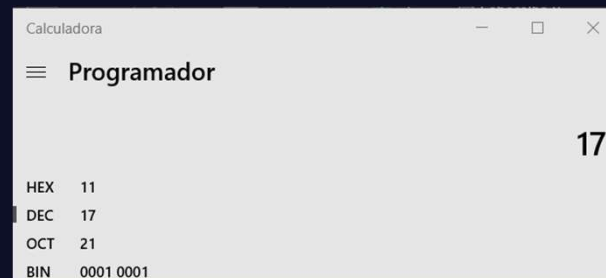
E com isso temos as variáveis conforme quadro ao lado

Valor das variáveis no final do ciclo do laço de repetição

Valor_decimal	0
Valor_octal	21
Sequencia	100

VALIDANDO RESULTADOS

DECIMAL	BINARIO	OCTAL
17	00010001	21



CODIGO FONTE

```
//ENQUANTO FOR A OPCAO 1 EXECUTA NOVAMENTE, SE FOR DIFERENTE DE 1 SISTEMA É FINALIZADO
while (opcao2 = 1) {
    //MOSTRA O MENU PRINCIPAL
    printf("\n-----\n");
    printf("##BEM VINDO AO CONVERSOR DE BASES!##\n");
    printf("\nDigite a Opcao de conversao desejada\n");
    printf("\n1 - Binario para Decimal\n");
    printf("\n2 - Decimal para Binario\n");
    printf("\n3 - Decimal para Octal\n");
    printf("\n4 - Octal para Decimal\n");
    printf("\n0 - Sair\n");
    printf("\nResposta: ");
    scanf("%c", &opcao);
    getchar();

    //OPÇÃO 0 - FECHA O SISTEMA
    if (opcao == '0') {

    //OPÇÃO 1 - BINÁRIO -> DECIMAL
    else if (opcao == '1') {

    //OPÇÃO 2 - DECIMAL -> BINÁRIO
    else if (opcao == '2') {

    //OPÇÃO 3 - DECIMAL -> OCTAL
    else if (opcao == '3') {

    //OPÇÃO 4 - OCTAL -> DECIMAL
    else if (opcao == '4') {
    //OPÇÃO DESCONHECIDA
    else {
        printf("Opcao desconhecida(%c)\n", opcao);
    }
    }
}
```

LAÇO DE REPETICAO E CONDICIONAIS

Foi criado um Menu que será exibido para que o usuário selecione uma opção de conversão.

Após esse menu temos uma estrutura if com condicionais que direcionam qual função será utilizada para realizar o cálculo.

Uma das opções é destinada a finalizar a aplicação, a opção 0.

Para conseguir voltar ao menu principal ao finalizar uma conversão adicionamos um laço de repetição (while) que será repetido sempre que o valor da variável opcao2 for igual a 1.

CODIGO FONTE

```
//OPÇÃO 0 - FECHA O SISTEMA
if (opcao == '0') {
    printf("Clique em ENTER e o sistema sera finalizado\n");
    opcao2 = 2;
    exit(0);
}

//OPÇÃO 1 - BINÁRIO -> DECIMAL
else if (opcao == '1') {
    printf("Digite numero binario: ");
    scanf("%lld", &valor_binario);
    printf("[%lld] em binario = [%d] em decimal\n", valor_binario, converterBinDec(valor_binario));
}

//OPÇÃO 2 - DECIMAL -> BINÁRIO
else if (opcao == '2') {
    printf("Digite o numero decimal: ");
    scanf("%d", &valor_decimal);
    printf("[%d] em decimal = [%lld] em binario\n", valor_decimal, converterDecBin(valor_decimal));
}

//OPÇÃO 3 - DECIMAL -> OCTAL
else if (opcao == '3') {
    printf("Digite o numero decimal: ");
    scanf("%d", &valor_decimal);
    printf("[%d] em decimal = [%d] em octal\n", valor_decimal, converterDecOct(valor_decimal));
}

//OPÇÃO 4 - OCTAL -> DECIMAL
else if (opcao == '4') {
    printf("Digite o numero octal: ");
    scanf("%d", &valor_octal);
    printf("[%d] em octal = [%lld] em decimal\n", valor_octal, converterOctDec(valor_octal));
}

//OPÇÃO DESCONHECIDA
else {
    printf("Opcao desconhecida [%c]\n", opcao);
}
}
```

LAÇO DE REPETICAO E CONDICIONAIS

Cada condicional (if, else if) possui uma mensagem que será impressa na tela perguntando qual o valor que deseja converter. Em seguida a aplicação salva o valor digitado em uma variável (scanf). E por fim exibe na tela uma mensagem o resultado. A opção 0 apenas exibe a mensagem "Clique em ENTER e o sistema será finalizado", pois essa opção é apenas para isso, fechar a aplicação.