|  |  |
| --- | --- |
|  | **UNIVERSIDADE DO ESTADO DE**  **SANTA CATARINA - UDESC**  **CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT** |

**Unidade Curricular:** Algoritmos

**Curso:** Bacharelado em Ciência da Computação

**Nome:** Gabriela Inácio

### **REVISÃO ESTRURAS DE REPETIÇÃO**

#### - Resolver 20 exercícios (do 4.5 em diante) para entregar, individual, arquivo .doc com [enunciado](https://moodle.joinville.udesc.br/mod/resource/view.php?id=227667) copiado. Anexar tela de dados de entrada e dados de saída no documento. Conta como nota complementar de trabalho.

**4.5 Ache o erro em cada um dos seguintes trechos de códigos. (Nota: pode haver mais de um erro.**

a) For ( x = 100**,** x >= 1**,** x++ )

printf( “%d\n”, x );

- For deve ser escrito em minusculo

-Dentro do parênteses do `for` está separado por vírgula, o correto seria por “;”

-Falta “{” depois dos parênteses e também no final do trecho de código “{“

-Opção de correção:

for ( x = 100;x >= 1;x++ ){

printf( “%d\n”, x );

}

b) O código a seguir deveria imprimir se determinado inteiro é ímpar ou par:

switch ( valor % 2 ) {

case 0:

printf( “Inteiro par\n” );

case 1:

printf( “Inteiro ímpar\n” );

}

* O trecho não tem uma condição para verificar se é impar ou par, ele apenas imprime na tela inteiro impar ou inteiro par
* Opção de correção:

int resto = valor % 2;

switch (resto) {

case 0:

if (resto == 0) {

printf("Inteiro par\n");

}

Break;

case 1:

if (resto != 0) {

printf("Inteiro ímpar\n");

}

Break;

}

c) O código a seguir deveria ler um inteiro e um caractere e imprimi-los. Suponha que o usuário digite 100 A.

scanf( “%d”, &intVal );

charVal = getchar();

printf( “Inteiro: %d\nCaractere: %c\n”,

intVal, charVal );

* O scanf vai ler o inteiro, sobrando “\n” e “A”
* Quando chama “charVal = getchar();” é lido apenas o “\n” fazendo com que o caractere não seja lido
* Opção de correção:

scanf( “%d”, &intVal );

getchar();

charVal = getchar();

printf( “Inteiro: %d\nCaractere: %c\n”, intVal, charVal );

d) for ( x = .000001; x == .0001; x += .000001 ) {

printf( “%.7f\n”, x );

}

* Precisa colocar um “0” antes do ponto
* A condição de parada é invalida
* Opção de correção:

for ( x = 0.000001; x <= 0.0001; x += 0.000001 ) {

printf( “%.7f\n”, x );

}

e) O código a seguir deveria exibir os inteiros ímpares de 999 a 1:

for ( x = 999; x >= 1; x += 2 ) {

printf( “%d\n”, x );

}

- Como a sequencia de numeros tem que decair, o incremento no looping `for` deveria ter decair e acrescentar.

- Opção de correção:

for ( x = 999; x >= 1; x -= 2 ) {

printf( “%d\n”, x );

}

f) O código a seguir deveria exibir os inteiros pares de 2 a 100:

contador = 2;

Do {

if ( contador % 2 == 0 ) {

printf( “%d\n”, contador );

}

contador += 2;

} While ( contador < 100 );

- Do deve ser escrito em minusculo

- While deve ser escrito em minusculo

-Não é necessário a condição imposta pelo if dentro do loop pois o contador inicia com 2 e é acrescentado 2 em 2

- para que considere o 100 também a condição do while deve ser “ contador <=100)

g) O código a seguir deveria somar os inteiros de 100 a 150 (supondo que total seja inicializado com 0):

for ( x = 100; x <= 150; x++ ); {

total += x;

}

* Tem um “;” depois do parêntesis que deve ser excluído

**4.6 Indique quais valores da variável de controle x são impressos por cada uma das seguintes estruturas for:**

a) for ( x = 2; x <= 13; x += 2 ) {

printf( “%d\n”, x );

}

R: 4, 6, 8, 10, 12

b) for ( x = 5; x <= 22; x += 7 ) {

printf( “%d\n”, x );

}

R: 12 e 19

c) for ( x =3 ; x <=15; x +=3 ) {

printf( “%d\n”, x );

}

R: 6, 9, 12, 15

d) for ( x = 1 ; x <=5; x +=7 ) {

printf( “%d\n”, x );

}

R: Não haverá retorno

e) for ( x =12 ; x >=2; x -=3 ) {

printf( “%d\n”, x );

}

R: 9, 6, 3

**4.7 Escreva estruturas for que imprimam a seguinte sequência de valores:**

a) 1,2,3,4,5,6,7

for ( x =0; x <=7; x +=1 ) {

printf( “%d\n”, x );

}

b)3,8,13,18,23

for ( x =-2; x <=23; x +=5 ) {

printf( “%d\n”, x );

}

c)20,14,8,2,-4,-10

for ( x =26; x >=-10; x -=6 ) {

printf( “%d\n”, x );

}

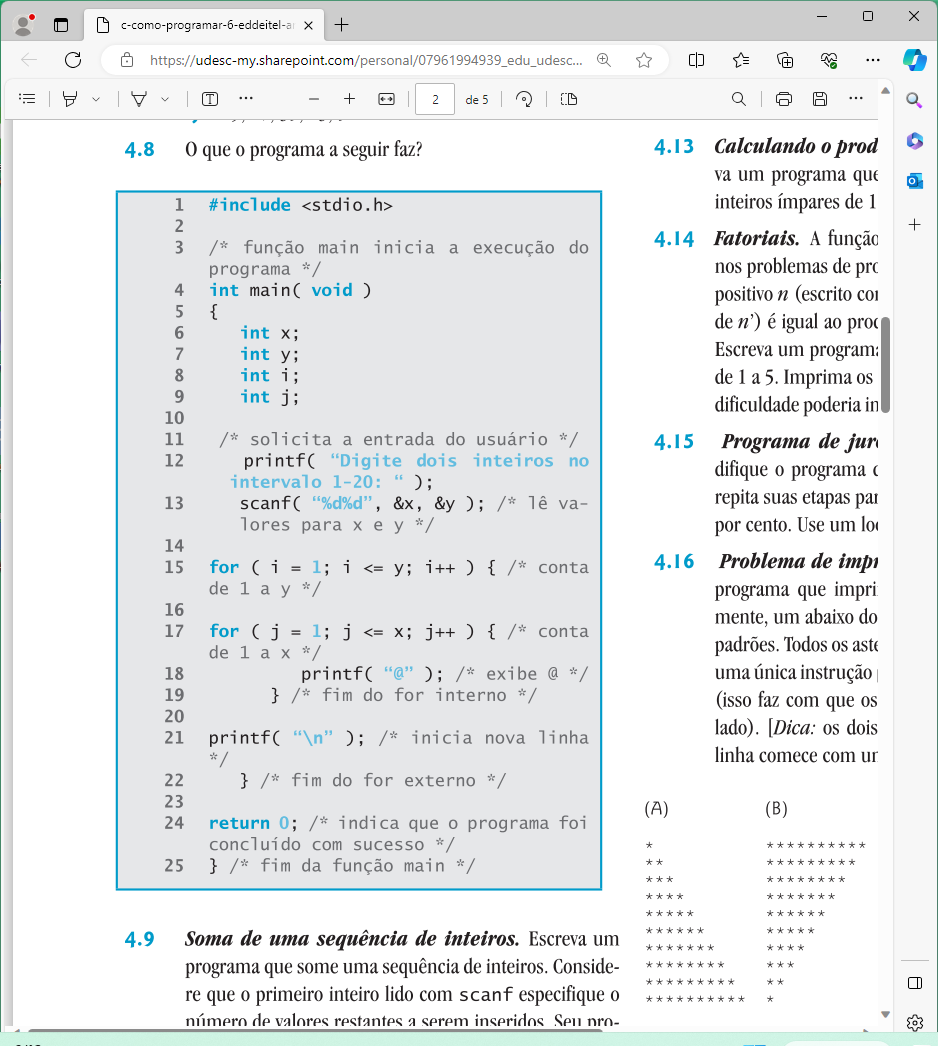
d)19,27,35,43,51

for ( x =26; x >=-10; x -=6 ) {

printf( “%d\n”, x );

}

**4.8 O que o programa a seguir faz?**



* O código solicita que o usuario insira dois numeros de 1 a 20. O primeiro numero que o usuario digitar (x) será a quantidade de colunas e o segundo (y) a quantidade de linhas, da tabela, composta por “@”, impressa na tela.

**4.9 *Soma de uma sequência de inteiros.*** Escreva um programa que some uma sequência de inteiros. Considere que o primeiro inteiro lido com *scanf*  especifique o número de valores restantes a serem inseridos. Seu programa deve ler apenas um valor toda vez que *scanf* for executado. Uma sequência típica poderia ser

5 100 200 300 400 500

Onde o 5 indica que os cinco valores subsequentes devem ser somados.

#include <stdio.h>

int main() {

int tam, i, soma = 0;

scanf("%d", &tam);

int vetor[tam];

for (i = 0; i < tam; i++) {

scanf("%i", &vetor[i]);

soma += vetor[i];

}

soma+=tam;

printf("%i", soma);

return 0;

}



**4.10 *Média de uma sequência de inteiros.*** Escreva um programa que calcule e imprima a média de vários inteiros. Considere que o último valor lido com scanf seja a sentinela 9999. Uma sequência de entrada típica poderia ser

10 8 11 7 9 9999

Indicando que é preciso calcular a média de todos os valores anteriores a 9999.

#include <stdio.h>

int main() {

int valor;

int i, soma = 0;

float media = 0;

for (i = 0; i < 9999; i++) {

scanf("%i", &valor);

if (valor == 9999) {

break;

} else {

soma += valor;

}

}

media = (float)soma / i;

printf("média: %.2f\n", media);

return 0;

}



**4.11 *Ache o menor.*** Escreva um programa que encontre o menor de vários inteiros. Considere que o primeiro valor lido especifique o número de valores restantes.

#include <stdio.h>

int main() {

int tam, i, menor;

printf("digite o primeiro numero: ");

scanf("%d", &tam);

int vetor[tam];

printf("digite o primeiro numero novamente: ");

scanf("%i", &vetor[0]);

menor = vetor[0];

for (i = 1; i < tam; i++) {

scanf("%i", &vetor[i]);

if (vetor[i] < menor) {

menor = vetor[i];

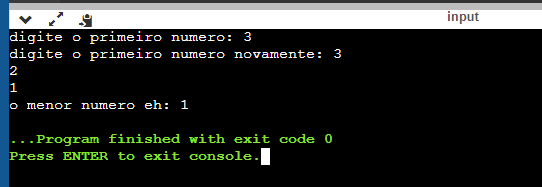
}

}

printf("o menor numero eh: %i", menor);

return 0;

}



**4.12 *Calculando a soma de inteiros pares.*** Escreva um programa que calcule e imprima a soma dos inteiros pares de 2 a 30

#include <stdio.h>

int main() {

int i, soma = 0;

for (i = 2; i <= 30; i += 2) {

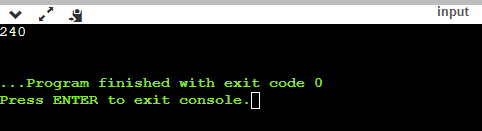
soma += i;

}

printf("%i\n", soma);

return 0;

}



**4.13 *Calculando o produto de inteiros ímpares.*** Escreva um programa que calcule e imprima o produto dos inteiros ímpares de 1 a 15.

#include <stdio.h>

int main() {

int valores[15];

int i, produto = 1;

for (i = 0; i < 15; i++) {

valores[i] = i + 1;

}

for (i = 0; i < 15; i++) {

if (valores[i] % 2 != 0) {

produto \*= valores[i];

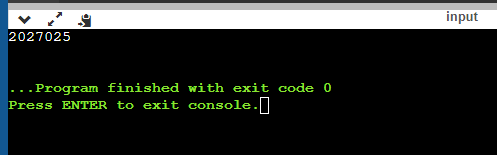
}

}

printf("%d\n", produto);

return 0;

}



**4.14 *Fatoriais.*** A função  *fatorial* é usada com frequência nos problemas de probabilidade. O fatorial de um inteiro positivo n (escrito como n! E pronunciado como ‘fatorial de n’) é igual ao produto dos inteiros positivos de 1 a n. Escreva um programa que avalie os fatoriais dos inteiros de 1 a 5. Imprima os resultados em formato tabular. Que dificuldade poderia impedi-lo de calcular o fatorial de 20?

#include <stdio.h>

int main() {

int valores[5];

int i, j, fatorial;

printf(" | fatorial |\n");

printf("--------------------------\n");

for (i = 0; i < 5; i++) {

valores[i] = i + 1;

fatorial = 1;

for (j = 1; j <= valores[i]; j++) {

fatorial \*= j;

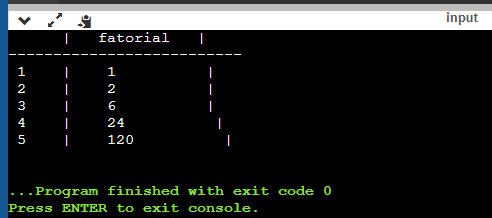
}

printf(" %d | %d |\n", valores[i], fatorial);

}

return 0;

}



Conforme aumenta-se o número n (número que deseja saber o fatorial) o resultado aumenta cada vez mais. Por exemplo, o fatorial de 20 (20!) resulta em 2.432.902.008.176.640.000. Tendo em vista que este é um numero muito grande, pode ser que haja problemas quando utilizá-lo como um tipo inteiro.

**4.15 *Programa de juros compostos modificado.*** Modifique o programa de juros compostos da seção 4.6 e repita suas etapas para taxas de juros de 5, 6, 7, 8, 9 e 10 por cento. Use um *loop for* para variar a taxa de juros.

#include <stdio.h>

#include <math.h>

int main() {

double valor;

double principal = 1000.0;

double taxa;

int ano;

printf("%4s%21s\n", "Ano", " Valor na conta");

for (taxa = 0.05; taxa <= 0.10; taxa += 0.01) {

printf("Taxa de juros: %.2f%%\n", taxa \* 100);

for (ano = 1; ano <= 10; ano++) {

valor = principal \* pow(1.0 + taxa, ano);

printf("%4d%21.2f\n", ano, valor);

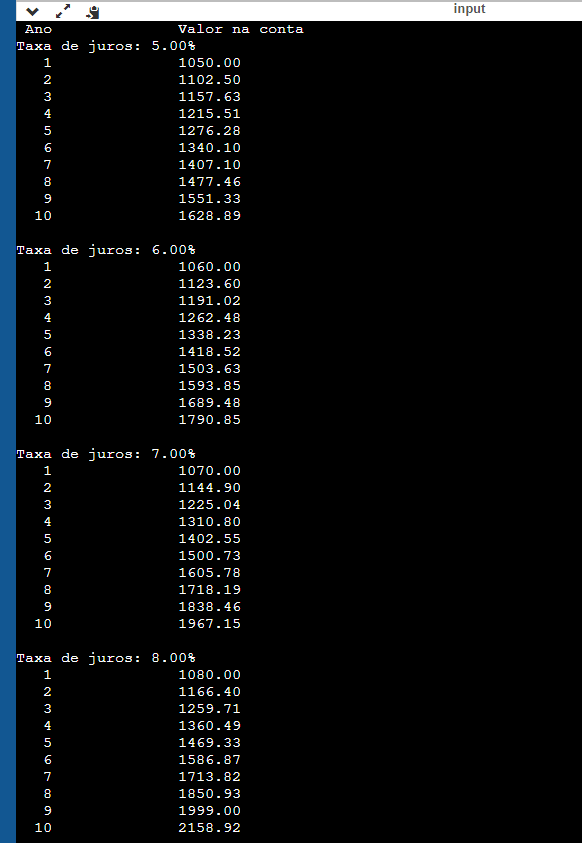
}

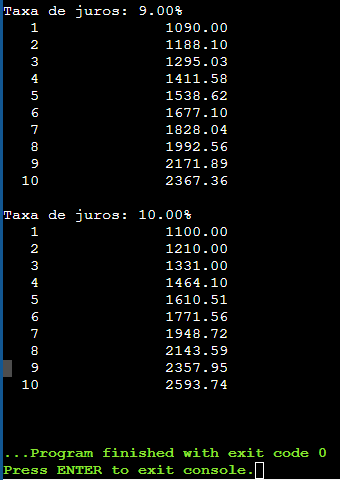
printf("\n");

}

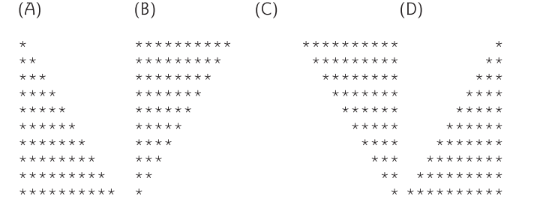
return 0;

}





**4.16** ***Problema de impressão de triângulo.*** Escreva um programa que imprima os padrões a seguir separadamente, um abaixo do outro. Use *loops for* para gerar os padrões. Todos os asteriscos (\*) devem ser impressos por uma única instrução *printf* na forma *printf(“\*”)*; (isso faz com que os asteriscos sejam impressos lado a lado). [*Dica:* os dois últimos padrões exigem que cada linha comece com um número de espaços apropriado.]



#include <stdio.h>

int main() {

int i, j,k;

printf("(A):\n");

for (i = 1; i <= 5; i++) {

for (j = 1; j <= i; j++) {

printf("\* ");

}

printf("\n");

}

printf("\n(B):\n");

for (i = 5; i >= 1; i--) {

for (j = 1; j <= i; j++) {

printf("\* ");

}

printf("\n");

}

printf("\n(C):\n");

for (i = 1; i <= 5; i++) {

for (j = 1; j <= 5 - i; j++) {

printf(" ");

}

for (k = 1; k <= i; k++) {

printf("\* ");

}

printf("\n");

}

printf("\(D):\n");

for (i = 5; i >= 1; i--) {

for (j = 1; j <= 5 - i; j++) {

printf(" ");

}

for (int k = 1; k <= i; k++) {

printf("\* ");

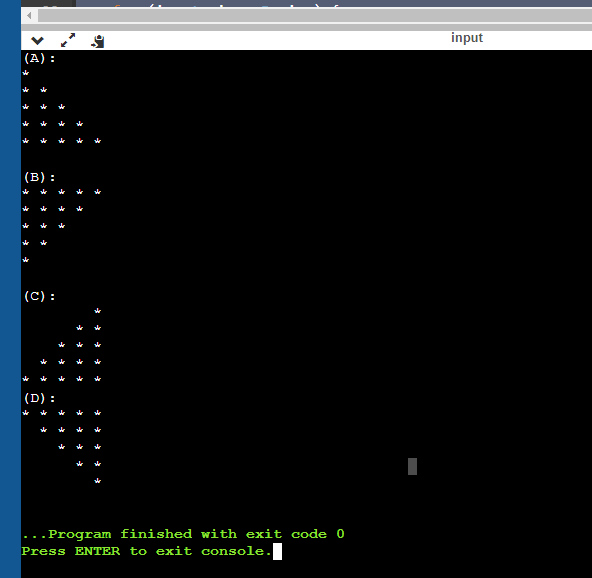
}

printf("\n");

}

return 0;

}



**4.17** ***Calculando limites de crédito.*** Poupar dinheiro vem se tornando algo cada vez mais difícil de se fazer durante períodos de recessão, de modo que as empresas podem estreitar seus limites de crédito para impedir que suas contas a receber (dinheiro devido a elas) se tornem muito grandes. Em resposta a uma recessão prolongada, uma empresa cortou os limites de crédito de seus clientes pela metade. Assim, se um cliente em particular tinha um limite de crédito de R$ 2.000,00, agora ele é de R$ 1.000,00. Se um cliente tinha um limite de R$ 5.000,00, agora ele é de R$ 2.500,00. Escreva um programa que analise o status de crédito de três clientes dessa empresa. Você receberá as seguintes informações:

**a)** O número da conta do cliente.

**b)** O limite de crédito do cliente antes da recessão.

**c)** O saldo atual do cliente (ou seja, o valor que o cliente deve à empresa).

Seu programa deve calcular e imprimir o novo limite de crédito para cada cliente e deve determinar (e imprimir) quais clientes têm saldo atual superior a seus novos limites de crédito.

#include <stdio.h>

int main() {

int conta[3];

double limiteantes[3], saldo[3], novolimite[3];

for (int i = 0; i < 3; i++) {

printf("\nCliente %i\n", i + 1);

printf("Número da conta do cliente:\n");

scanf("%i", &conta[i]);

printf("Limite de crédito do cliente antes da recessão: ");

scanf("%lf", &limiteantes[i]);

printf("Saldo atual do cliente (ou seja, o valor que o cliente deve à empresa):\n");

scanf("%lf", &saldo[i]);

novolimite[i] = limiteantes[i] / 2;

}

printf("\nNumero da conta | Novo limite de credito\n");

printf("----------------|-----------------------\n");

for (int i = 0; i < 3; i++) {

printf("%15i | R$ %.2lf\n", conta[i], novolimite[i]);

if (saldo[i] > novolimite[i]) {

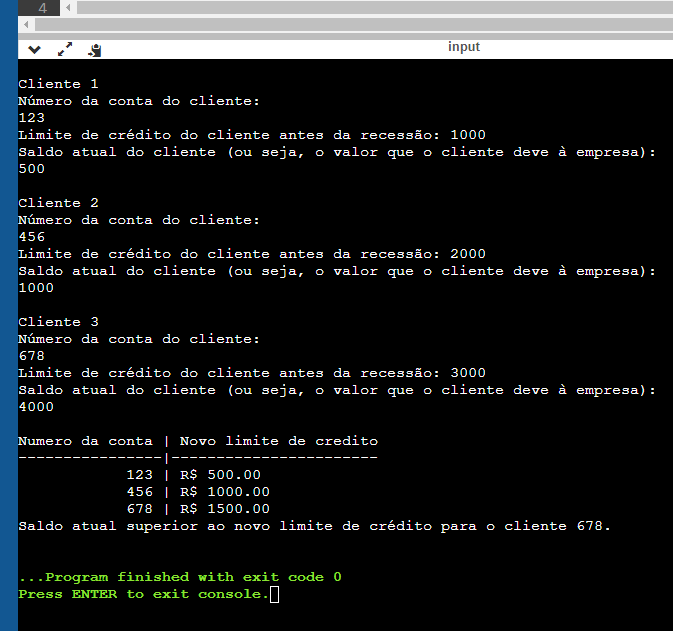
printf("Saldo atual superior ao novo limite de crédito para o cliente %i.\n", conta[i]);

}

}

return 0;

}



**4.18** ***Programa de exibição de gráficos de barras.*** Uma aplicação interessante dos computadores é a de desenhar gráficos e gráficos de barras (às vezes, chamados ‘histogramas’). Escreva um programa que leia cinco números (entre 1 e 30). Para cada número lido, seu programa deverá exibir uma linha contendo esse número de asteriscos adjacentes. Por exemplo, se seu programa ler o número sete, ele deverá exibir \*\*\*\*\*\*\*.

#include <stdio.h>

int main() {

int vetor[5], i, j;

for (i = 0; i < 5; i++) {

printf("Digite um numero entre 1-30:\n");

for (;;) {

printf("numero %d: ", i + 1);

scanf("%d", &vetor[i]);

if (vetor[i] >= 1 && vetor[i] <= 30) {

break;

}

}

}

for (i = 0; i < 5; i++) {

printf("%2d: ", vetor[i]);

for (j = 0; j < vetor[i]; j++) {

printf("\*");

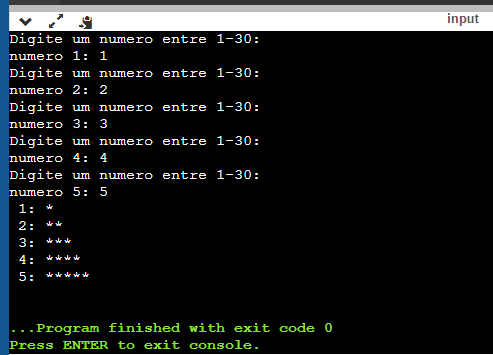
}

printf("\n");

}

return 0;

}



**4.19 *Calculando vendas.*** Um varejista on-line vende cinco produtos diferentes cujos preços de revenda aparecem na tabela a seguir:



Escreva um programa que leia uma série de pares de números da seguinte forma:

1. Número do produto.
2. Quantidade vendida durante um dia.

Seu programa deverá usar uma estrutura *switch* para ajudar a determinar o preço de revenda para cada produto. O programa deverá calcular e exibir o valor de revenda total de todos os produtos vendidos na semana anterior.

#include <stdio.h>

int main() {

int produto, quantidade;

float preco, total = 0;

while (1) {

printf("Digite o número do produto, de 1-5, ou -1 para encerrar: \n");

scanf("%d", &produto);

if (produto == -1) {

break;

}

switch (produto) {

case 1:

preco = 2.98;

break;

case 2:

preco = 4.50;

break;

case 3:

preco = 9.98;

break;

case 4:

preco = 4.49;

break;

case 5:

preco = 6.87;

break;

}

printf("Qual a quantidade que foi vendida?\n ");

scanf("%d", &quantidade);

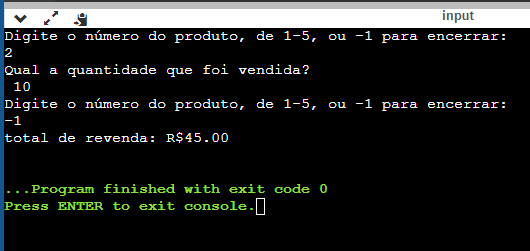
total += preco \* quantidade;

}

printf("total de revenda: R$%.2f\n", total);

return 0;

}



**4.20 *Tabelas verdade.*** Complete as tabelas verdade a seguir preenchendo cada espaço com 0 ou 1.

Resposta em vermelho na imagem

**4.21** Reescreva o programa da Figura 4.2 de modo que a inicialização da variável *contador* seja feita na declaração e não na estrutura *for.*

#include <stdio.h>

int main() {

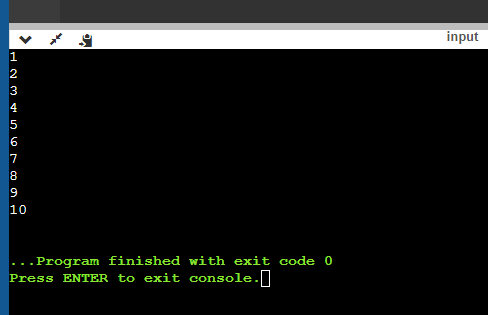
for (int contador = 1; contador <= 10; contador++) {

printf("%d\n", contador);

}

return 0;

}



**4.23 *Calculando juros compostos com inteiros.*** Modifique o programa da Figura 4.6 de modo que ele use apenas inteiros para calcular os juros compostos. [*Dica:* trate todos os valores monetários como números inteiros de centavos. Depois ‘quebre’ o resultado em sua parte de real e em sua parte de centavos usando as operações de divisão e módulo, respectivamente. Insira uma vírgula.]

#include <stdio.h>

#include <math.h>

int main(void) {

int centavos, principal = 100000, taxa = 5, ano, reais;

printf("%4s%21s\n", "Ano", "Valor na conta");

for (ano = 1; ano <= 10; ano++) {

centavos = principal \* pow(1.0 + taxa / 100.0, ano);

reais = centavos / 100;

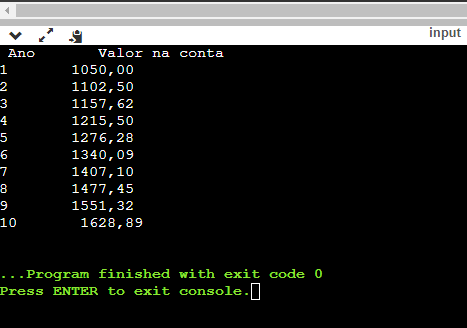
centavos = centavos % 100;

printf("%i%11i,%02i\n", ano, reais, centavos);

}

return 0;

}



**4.24** *Considere i = 1, j = 2, k = 3 e m = 2.* O que cada uma das seguintes instruções imprime?

a) printf( “%d”, i == 1 );

- Imprime 1

b) printf( “%d”, j == 3 );

- Imprime 0

c) printf( “%d”, i >= 1 && j < 4 );

- Imprime 1

d) printf( “%d”, m < = 99 && k < m );

- Imprime 1

e) printf( “%d”, j >= i || k == m );

- Imprime 1

f) printf( “%d”, k + m < j || 3 - j >= k );

- Imprime 0

g) printf( “%d”, !m );

- Imprime 0

h) printf( “%d”, !( j - m ) );

- Imprime 1

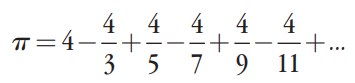
i) printf( “%d”, !( k > m ) );

- Imprime 0

j) printf( “%d”, !( j > k ) );

- Imprime 1

***4.26 Calculando o valor de pi .*** Calcule o valor de p a partir da série infinita



imprima uma tabela que mostre o valor de pi aproximado por um termo dessa série, e depois por dois termos, três termos, e assim por diante. Quantos termos dessa série você precisa usar antes de obter 3,14? 3,141? 3,1415? 3,14159?

#include <stdio.h>

int main() {

int termos;

double pi = 0.0;

printf("quantos termos a serie terá? ");

scanf("%d", &termos);

printf("\ntermos\t pi\n");

for (int i = 0; i < termos; i++) {

if (i % 2 == 0) {

pi += 4.0 / (2 \* i + 1);

} else {

pi -= 4.0 / (2 \* i + 1);

}

printf("%d\t%.10f\n", i + 1, pi);

}

return 0;

}

