Resolução exercícios (parte lógica):

* Implemente um código em C para aprovar empréstimo bancário. O código deve pedir as seguintes informações*: nome do cliente*, *idade do cliente*, valor *do empréstimo*, *número de parcelas e salário do cliente.* Como resultado, o programa deve apresentar *o nome do cliente, a idade, se o empréstimo foi aprovado, quantas parcelas e o valor da parcela,* levando em consideração as seguintes diretrizes:

- O empréstimo só será aprovado se:

---O valor das parcelas do empréstimo represente no máximo 30% do salário do solicitante.

---A última parcela for paga antes do cliente completar 50 anos de idade.

-Para calcular o valor da parcela:

---deve-se aplicar 23% de juros no valor total do empréstimo antes de fazer a divisão pela quantidade de parcelas, no caso do total de parcelas ser maior ou igual a 300.

---deve-se aplicar 17% de juros no valor total do empréstimo antes de fazer a divisão pela quantidade de parcelas, no caso do total de parcelas ser maior ou igual a 200 e menor que 300.

---deve-se aplicar 9% de juros no valor total do empréstimo antes de fazer a divisão pela quantidade de parcelas, no caso do total de parcelas ser maior ou igual a 50 e menor que 100.

---deve-se aplicar 5% de juros no valor total do empréstimo antes de fazer a divisão pela quantidade de parcelas, no caso do total de parcelas ser menor que 50.

Interpretação:

-Variáveis que vão ser solicitadas para o usuário: *nome do cliente*, *idade do cliente*, valor *do empréstimo*, *número de parcelas e salário do cliente.*

-Saídas: *o nome do cliente, a idade, se o empréstimo foi aprovado, quantas parcelas e o valor da parcela*

-Regras para aprovar:

valorParcela <= 30% do salario

idade\_ultima\_parcela = quantidade de parcelas + idade cliente

idade\_ultima\_parcela < 50

-Regras para calcular parcela:

Se total\_parcelas >= 300 faça valor\_total\_emprestimo + 23%

Se total\_parcelas >=200 e total\_parcelas <300 faça valor\_total\_emprestimo + 17%

Se total\_parcelas >=50 e total\_parcelas <100 faça valor\_total\_emprestimo + 9%

Se total\_parcelas <50 faça valor\_total\_emprestimo + 5%

Resultado final:

-------------------------------------------------------

-Nome do cliente: \*\*\*\*\*\*

-Idade do cliente: \*\*\*\*

-Empréstimo aprovado ou não : \*\*\*\*

-Numero de parcelas : \*\*\*

-Valor da parcela: \*\*\*

-------------------------------------------------------

* Uma transportadora de combustível avalia mensalmente seus funcionários de acordo com o tempo de rodagem, consumo de combustível, recebimento de multas e avaliação dos clientes (avaliação de 0 a 10). Para realizar essa avaliação, a empresa pega *2 dias aleatórios de um mês que será digitado pelo usuário, um mês* *e um ano*, e para **cada um destes dias**, recebe a *distância em km da base ao cliente de destino, o consumo de combustível, a hora de saída (somente horas), a hora de chegada no cliente, o código do funcionário, o salário do funcionário, a avaliação do cliente que recebeu o produto e se o motorista recebeu multa nesse período ou não.*

Com esses dados em mãos, a empresa **calcula** o tempo total de rodagem, com a soma da ida e volta até o cliente, o tempo total de rodagem, e a velocidade média de cada dia.

Neste sentido, desenvolva um programa em C que receba todos estes dados iniciais que a empresa recebe, faça os cálculos citados, e em seguida, calcule a bonificação que o motorista irá receber, e apresente na tela, o código do funcionário, o salário inicial, o valor da bonificação e o salário após a soma da bonificação utilizando as seguintes diretrizes:

--caso o motorista não tenha recebido multa, tenha trabalhado no mínimo 10hrs por dia nos 2 dias, e feito uma média de combustível por dia, maior que 6km/l, e não ter recebido avaliação menor que 6, deve-se aplicar 5% de bonificação sobre o salário mensal do funcionário.

--caso o motorista tenha recebido multa, mas tenha trabalho no mínimo 12hrs por dia, nos 2 dias, e feito uma média de combustível por dia, maior que 10km/l, e não ter recebido avaliação menor que 6, deve-se aplicar 6% de bonificação sobre o salário mensal do funcionário.

----caso o motorista tenha recebido multa, mas tenha trabalho no mínimo 12hrs em um dia e no mínimo 8hrs em outro dia, e feito uma média de combustível por dia, maior que 7km/l, e não ter recebido avaliação menor que 6, deve-se aplicar 4.5% de bonificação sobre o salário mensal do funcionário.

--caso o motorista não tenha recebido multa, tenha trabalho no mínimo 10hrs por dia, nos 2 dias, e feito uma média de combustível por dia, maior que 7km/l, e não ter recebido avaliação menor que 8, deve-se aplicar 9% de bonificação sobre o salário mensal do funcionário.

Interpretação:

Variáveis que vão ser solicitadas pelo usuário:

*dia1, dia2, mês, ano, distância\_base\_cliente1, distancia\_base\_cliente2, consumo\_combustivel1, consumo\_combustivel2, hora\_saida1, horasaida2, hora\_chegada1, hora\_chegada2, código\_funcionario, salario\_funcionario, nota1, nota2, multa1, multa2*

Cálculos da empresa:

Tempo\_total\_rodagem1: hora\_chegada1– hora\_saida1

Tempo\_total\_rodagem2: hora\_chegada2– hora\_saida2

Velocidade\_media1: distancia\_base\_cliente1 / tempo\_total-rodagem1

Velocidade\_media2: distancia\_base\_cliente2 / tempo\_total-rodagem2

REVISÃO PARA PROVA

1. Qual a função que está presente em todos os programas em C?

A função “main”

1. Como devem terminar todas as instruções em C?

É necessário terminar com “;”

1. Para que serve a linha **#include<stdio.h>**?

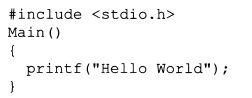
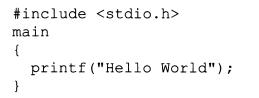
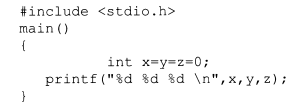
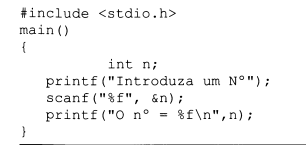
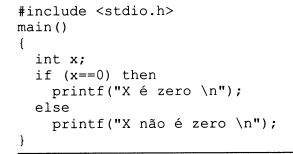
Para importar a biblioteca “stdio.h”

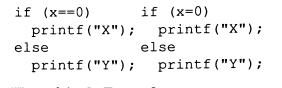
1. O que é um pensamento computacional?

A forma que um computador vai ler um determinado conjunto de comandos.

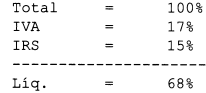
1. O que é um pseudocódigo e para que serve?

Uma forma de demonstrar a lógica de programação, no qual pode ser aplicada em diversas linguagens de programação.

1. Marque as alternativas verdadeiras e justifique as falsas.
   1. O *else* é um *if* facultativo. (Verdadeira)
   2. Num *if* são necessário parênteses em torno da condição. (Verdadeira)
   3. Tanto o *if* quanto o *else*  só podem podem conter uma única condição. (Falso, um if pode haver várias condições, desde que estejam coerentes)
   4. Tanto o *if*  quanto o *else*  só podem podem conter uma única instrução. (Falso, pode haver quantas instruções que o programador desejar, não havendo limite.
2. Marque a alternativa correta:
   1. Os comentários devem ser inscritos antes de qualquer instrução do programa.
   2. Os comentários devem ser sempre depois das instruções.
   3. Os comentários devem ser antes do main
   4. Os comentários podem ser colocados sempre que o programador quiser.
3. Marque as alternativas que contém declaração de variável válida.
   1. Y int;
   2. int;
   3. integer x;
   4. inta,b;
   5. float c, int a;
   6. float a, b, c, d;
   7. char = 1
4. Identifique os erros que serão detectados ao executar os programas a seguir.
   1. O “main” está em maiúsculo, logo o código não é válido. Pois a linguagem C é Case Sensitive.
   2. Não há “()” após o “main”, logo o código não será executado.
   3. Não é possível declarar todas a variáveis ao menos tempo e declarar um valor para os mesmos.
   4.  O erro está no %f dentro do scanf e do printf, pois o corretor seria %d, já que os valores que serão guardados e inseridos são inteiros.
   5. Não é possível utilizar “then” após o if para indicar o que acontecera, o correto seria abrir o escopo com “{}” para múltiplas instruções ou somente colocar uma instrução após o if.
5. Qual a diferença entre os seguintes trechos de código.

O primeiro código no if há uma comparação, porém no segundo não, já que está tornando o valor de X para 0, então não importa o resultado, sempre cairá no else.

1. Escreva um programa que coloque na tela a seguinte saída:



#include <stdio.h>

int main(){

printf("Total = 100%%\n");

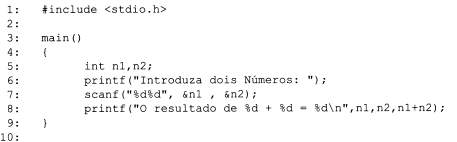
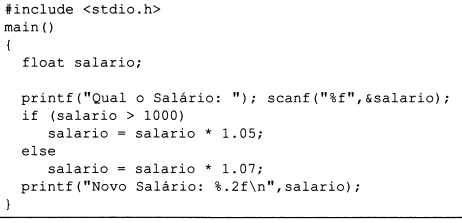
printf("IVA = 17%%\n");

printf("IRS = 15%%\n");

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

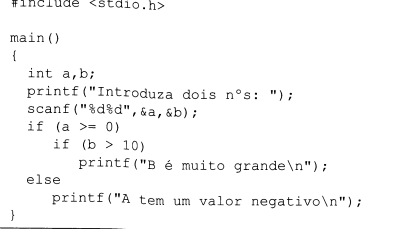
printf("Liq. = 68%%\n");

}

1. Escreva as saídas dos programas a seguir
   1. ”O “O resultado de 4 + 5 = 9” (Valores inseridos 4 e 5)
   2. 

Valor exemplo: 900: Novo salario: 945.00

Valor exemplo: 1050: Novo Salário: 1123.50

* 1. 

Valor exemplo: 4 e 11: “B é muito grande”

Valor exemplo: -5 e 50: “A tem um valor negativo”

1. Implemente um programa que solicite ao usuário dois inteiros e em seguida aplique todos os operadores relacionais em C ( ==,>,<,>=,<=,!=)

#include <stdio.h>

int main(){

int x,

y;

printf("Insira os dois valores: ");

scanf("%d%d", &x, &y);

if (x == y){

printf("%d == %d\n", x, y);

}

if (x > y){

printf("%d > %d\n", x, y);

}

if (x < y){

printf("%d < %d\n", x, y);

}

if (x >= y){

printf("%d >= %d\n", x, y);

}

if (x <= y){

printf("%d <= %d\n", x, y);

}

if (x != y){

printf("%d != %d\n", x, y);

}

}

1. Implemente um programa que adicione mil reais aos salario de um indivíduo, caso o salario do mesmo seja inferior a cem mil.

#include <stdio.h>

int main(){

float

sal;

printf("Informe o salario do individuo: ");

scanf("%f", &sal);

if (sal < 100000){

sal += 1000;

}

printf("O salario atual e: %.2f", sal);

}

1. Escreva um programa que indique quantos segundos tem um determinado número de horas.

#include <stdio.h>

int main(){

int

h;

printf("Informe a quantida de horas: ");

scanf("%d", &h);

printf("%d horas em segundos: %d segundos", h, (h\*60)\*60);

}

1. Escreva um programa que verifique se o número é bissexto ou não.

#include <stdio.h>

int main(){

int

i;

printf("Informe um numero: ");

scanf("%d", &i);

if (i%4 == 0){

printf("%d e um numero bissexto!", i);

}

else {

printf("%d nao e um numero bissexto!", i);

}

}