

Trabalho 1

LEIA ATENTAMENTE AS REGRAS E OS ENUNCIADOS

R E G R A S

- O trabalho deverá ser enviado para o Google Classroom até o dia 22/05/2023 (segunda).
- A data de entrega não será adiada.
- A prova poderá ser feita em dupla ou individualmente. Se for em dupla, todos os 2 integrantes devem realizar a entrega do mesmo arquivo dentro do prazo.
- Os 3 programas solicitados (arquivos **.c**) deverão ser compactados em um único arquivo (apenas **.ZIP**) com o nome e sobrenome do aluno, seguido da matrícula, conforme os exemplos abaixo.
 - Individual: **JoseBonifacioDeAndrade20212210999.zip**
 - Dupla: **AnaMaria20221210999_JoseBonifacio20212210999.zip**
- Não envie outros arquivos dentro do ZIP. Somente os arquivos com extensão **.c**.
 - *Dica: após compactar, verifique o conteúdo do arquivo compactado.*
- Os programas (arquivos **.c**) deverão ter os nomes conforme definido no início dos enunciados.
- Não será permitido o uso de variáveis globais.
- Não serão aceitos trabalhos enviados por e-mail.
- Trabalhos com estruturas e/ou organizações semelhantes (plágio) serão penalizados com a nota zero.
- Cada arquivo **.c** deve ser passível de compilação (usando o comando **gcc/g++**) e de posterior execução sem erros.
- O programa que não obedecer às restrições estabelecidas receberá zero.

ENUNCIADOS

1) Programa: arctan.c (2,0 pontos)

O valor aproximado do arco-tangente de x (para $|x| < 1$) é calculado pela seguinte série:

$$\arctan(x) = \sum_{i=0}^n \frac{(-1)^i}{2i+1} x^{2i+1}$$

Crie um programa em C que lê primeiramente um valor x ($|x| < 1$), e em seguida um valor n ($n \geq 0$) onde n é o número de termos da série e imprime o valor do arco-tangente, calculado de acordo com a fórmula acima.

Restrições:

- a) $|x| < 1$. Se $|x| \geq 1$, então solicite novamente o valor até o usuário digitar corretamente.
- b) $n \geq 0$. Se $n < 0$, então solicite novamente o valor até o usuário digitar corretamente.
- c) Não poderá ser usada nenhuma função matemática implementada na biblioteca do C/C++ nem em nenhuma outra biblioteca.
- d) Não poderão ser usados vetores nem qualquer outro tipo de estrutura de dados, somente variáveis simples.
- e) Todo o código deverá estar implementado na função main, sem o uso de funções auxiliares, com a exceção das funções **scanf**, **printf** e **fflush**.

Dica: se achar necessário, poderá usar tipos de dados com larga abrangência e/ou com alta precisão, como o **long int** ("%ld"), **long long int** ("%lld"), **double** ("%lf"), ou o **long double** ("%Lf").

Exemplo de execução (o formato das saídas e entradas deve seguir esse exemplo):

1

O valor absoluto de x deve ser menor que 1. Informe novamente.

0.5

-1

O valor de n deve ser maior ou igual a 0. Informe novamente.

0

0.5

2) Programa: caracoroa.c (3,0 pontos)

João e Maria são amigos desde que se conheceram na creche. Desde então, eles compartilham uma rotina de brincadeiras: todas as vezes que eles se encontram, eles jogam Cara ou Coroa com uma moeda, e quem ganhar tem o privilégio de decidir quais brincadeiras eles irão jogar durante o dia. Maria sempre escolhe cara, e João sempre escolhe coroa.

Hoje em dia eles estão na faculdade, mas continuam sendo bons amigos. Sempre que se encontram, eles ainda jogam Cara ou Coroa, e o vencedor decide que filme assistir, ou em que restaurante jantar, e assim por diante.

Ontem Maria contou a João que ela guarda um registro de todas as vezes que eles jogaram, desde os tempos da creche. João ficou espantado. Porém João está estudando Programação e decidiu que essa era uma boa oportunidade para mostrar a Maria suas habilidades em programação, escrevendo um programa que mostrasse o número de vezes que cada um deles venceu ao longo de todos esses anos.

Entrada: A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um único inteiro N indicando o número de vezes jogadas ($1 \leq N \leq 10000$). A linha seguinte contém N inteiros R_i , separados por um espaço, descrevendo a lista de resultados. Se $R_i = 0$ então Maria venceu o i -ésimo jogo, se $R_i = 1$ então João venceu o i -ésimo jogo ($1 \leq i \leq N$). O fim da entrada é indicado por $N = 0$.

Saída: Para cada caso de teste na entrada, seu programa deverá escrever uma linha contendo a sentença "Maria venceu X vez(es) e João venceu Y vez(es)", onde $0 \leq X$ e $0 \leq Y$.

Exemplo de execução (o formato das saídas e entradas deve seguir esse exemplo):

5

0 0 1 0 1

Maria venceu 3 vez(es) e Joao venceu 2 vez(es) .

6

0 0 0 0 0 1

Maria venceu 5 vez(es) e Joao venceu 1 vez(es) .

0

Escreva o programa em C, similar ao que o João pretende escrever.

Restrições:

- a) Se valor digitado do número de vezes jogadas for diferente do valor que indica o fim do programa, ou estiver fora da faixa de valores válidos, informe uma mensagem de erro, e peça para o usuário digitar novamente um valor válido.
- b) Se um valor digitado dentro da rodada for diferente de 0 ou de 1, informe uma mensagem de erro, e aborte a rodada.
- c) Para cada linha, após ler os valores 0's e 1's das jogadas de uma rodada, o restante do conteúdo da linha deve ser descartado.
- d) Não poderá ser usada nenhuma função matemática implementada na biblioteca do C/C++ nem em nenhuma outra biblioteca.
- e) Não poderão ser usados vetores nem qualquer outro tipo de estrutura de dados, somente variáveis simples.
- f) Todo o código deverá estar implementado na função **main**, sem o uso de funções auxiliares, com a exceção das funções **scanf**, **printf** e **fflush**.

3) Programa: salario.c (5,0 pontos)

Uma empresa de TI paga seus funcionários por hora trabalhada, de acordo com a função e a experiência, com base na tabela abaixo:

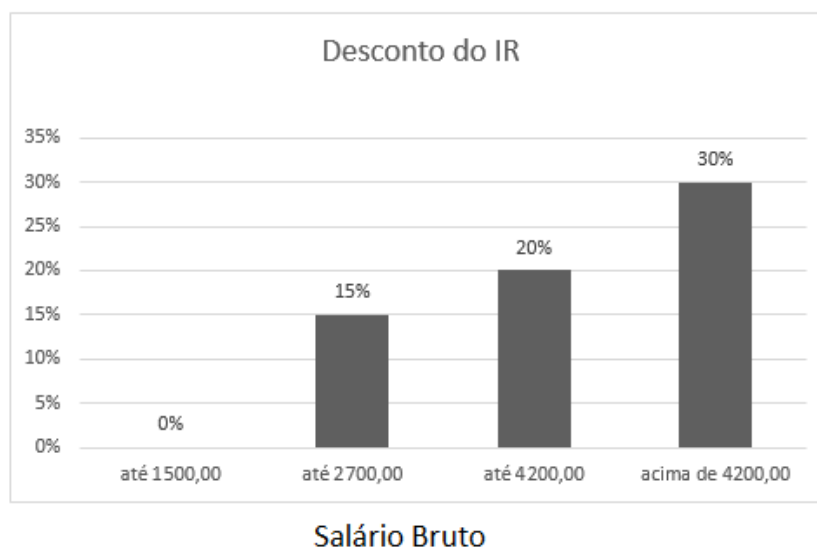
Experiência	Função		
	Programador	Analista	Gerente
Até 2 anos	25,00	45,00	85,00
De 3 a 5 anos	30,00	55,00	102,00
Mais de 5 anos	38,00	70,00	130,00

Além disso, cada funcionário tem em seu contrato de trabalho o valor total de horas contratadas no mês. Por exemplo, um programador pode ser contratado para trabalhar 80 horas/mês e um gerente pode ser contratado para trabalhar 120 horas/mês. Se o total de horas trabalhadas em um mês for maior que a quantidade de horas contratadas, o valor excedente deve ser pago como horas extras, de acordo com a seguinte tabela:

Horas excedentes	Percentual
Até 13 horas	23%
Mais de 13 horas até 22 horas	37%
Acima de 22 horas	56%

Por outro lado, se o total de horas trabalhadas em um mês for menor que a quantidade de horas contratadas, ele só receberá as horas trabalhadas.

Sobre o salário bruto incidem 2 descontos: INSS e IR. O desconto do INSS incide sobre o salário bruto e é de 11% para todos os funcionários. O desconto do IR incide sobre salário bruto – INSS e é calculado de acordo com o gráfico abaixo:



Crie um programa em C para ler a quantidade de funcionários (valor inteiro > 0) e, para cada funcionário, ler:

- Função do Funcionário (P-programador, A-analista ou G-gerente);
- Anos de experiencia (valor inteiro ≥ 0);
- Horas contratadas no mês (valor inteiro > 0);
- Horas trabalhadas no mês (valor inteiro ≥ 0).

Para cada funcionário deve-se calcular o salário e imprimir a folha de pagamento com as seguintes informações: o salário bruto no mês, as horas excedentes (se houver), os descontos do INSS e IR e o salário líquido.

Estruture o código, criando e utilizando as 3 sub-rotinas descritas abaixo:

- **lerFuncionario**: esta sub-rotina deve ler do usuário os dados de um funcionário, e devolver esses dados à rotina principal através de parâmetros de saída.
- **calcularSalario**: esta sub-rotina deve receber como parâmetros de entrada a função de um funcionário, a quantidade de anos de experiência desse funcionário, a quantidade de horas contratadas no mês desse funcionário, e a quantidade de horas trabalhadas no mês desse funcionário, e deve retornar o salário líquido desse mesmo funcionário. Além desse retorno e dos parâmetros de entrada, essa sub-rotina também deve conter os seguintes parâmetros de saída: o salário bruto no mês, a quantidade de horas excedentes, o desconto do INSS e o desconto do IR.
- **imprimirFolhaPagamento**: esta sub-rotina deve receber como parâmetros todos os dados pertinentes para que a folha de pagamento seja impressa: o salário bruto no mês, as horas excedentes, os descontos do INSS e IR e o salário líquido. Esta rotina deve imprimir os dados da folha de pagamento de um funcionário, e as horas excedentes só devem ser impressas se de fato houverem. Essa sub-rotina não deve retornar nada.

Exemplo de execução (o formato das saídas e entradas deve seguir esse exemplo):

Qtd: 0

ATENÇÃO: a quantidade de funcionários deve ser maior que zero. Informe novamente.

Qtd: 2

=====

Funcionário 1

- Função: P

- Anos de Exp.: 5

- Horas contratadas: 160

- Horas trabalhadas: 200

Folha de Pagamento do Func. 1

- Salário Bruto... (R\$): 6672,00

- Horas Excedentes (h): 40hr

- Desconto INSS... (R\$): 733,92
- Desconto IR.... (R\$): 1781,42
- Salário Líquido. (R\$): 4156,66

=====

Funcionário 2

- Função: **g**
- Anos de Exp.: **9**
- Horas contratadas: **120**
- Horas trabalhadas: **100**

Folha de Pagamento do Func. 2

- Salário Bruto... (R\$): 13000,00
- Desconto INSS... (R\$): 1430,00
- Desconto IR.... (R\$): 3471,00
- Salário Líquido. (R\$): 8099,00

Restrições:

- g) A função de cada funcionário P, A ou G pode ser fornecida em maiúscula ou minúscula.
- h) Caso algum dado fornecido não esteja correto, o programa deve apresentar uma mensagem de erro e lê-lo novamente até que o usuário digite corretamente.
- i) Não poderão ser usados agregados de dados (vetores, matrizes ou structs), somente variáveis simples.
- j) Todo o código deverá estar implementado ou rotina **main**, e nas sub-rotinas **lerFuncionario**, **calcularSalario** e **imprimirFolhaPagamento**, sem a criação ou uso de quaisquer outras rotinas auxiliares, com a exceção das funções **scanf**, **printf** e **fflush**, e das funções contidas na biblioteca matemática **math.h**, e da biblioteca **locale.h** para melhor representação dos caracteres.
- a) Não são permitidas variáveis globais. A definição de constantes é permitida.