

Trabalho 1

LEIA ATENTAMENTE AS REGRAS E OS ENUNCIADOS

R E G R A S

- O trabalho deverá ser realizado individualmente.
- O trabalho deverá ser enviado para o Google classroom até o dia 19/06/2021 (domingo).
- A data de entrega não será adiada.
- Os 4 programas solicitados (arquivos .CPP) deverão ser compactados em um único arquivo (ZIP ou RAR) com o nome e sobrenome do aluno.
- Deverão ser compactados somente os arquivos .CPP (não enviar arquivos executáveis!)
- Os programas (arquivos .CPP) deverão ter os nomes conforme definido nos enunciados.
- Não serão aceitos trabalhos enviados por email.
- Trabalhos com estruturas e/ou organizações semelhantes (plágio) serão penalizados com a nota zero.

1) Programa: lavoura.cpp (2,0 pontos)

Uma empresa utiliza aviões para pulverizar lavouras. O valor do serviço cobrado pela empresa depende da área a ser pulverizada e do tipo da pulverização, conforme tabela a seguir:

Tipo da pulverização	Valor
E - ervas daninhas	R\$ 50,00 / acre
G - gafanhotos	R\$ 100,00 / acre
B - broca	R\$ 150,00 / acre
A - ervas daninhas + gafanhotos	R\$ 140,00 / acre
F - ervas daninhas + broca	R\$ 190,00 / acre
H - gafanhotos + broca	R\$ 240,00 / acre
T - todos	R\$ 270,00 / acre

Se a área a ser pulverizada for superior a 10 acres, o fazendeiro tem um desconto de 5% sobre o valor do serviço. Além disso, se o valor do serviço for maior do que R\$ 750,00, o fazendeiro tem um desconto de 10% sobre o valor que ultrapassar os R\$ 750,00. Caso ambos os descontos se apliquem, o desconto da área é calculado primeiro.

Crie um programa em C para ler os dados da pulverização e imprimir o valor total do serviço, o valor do desconto (se houver) e o valor final a ser pago pelo fazendeiro.

Restrições:

- Quantidade de acres > 0 . Se valor digitado ≤ 0 , então solicite novamente o valor até o usuário digitar corretamente.
- O tipo da pulverização deve ser uma das opções da tabela (letra maiúscula ou minúscula). Se o valor digitado não corresponder a nenhum item da tabela, solicite novamente o valor até o usuário digitar corretamente.
- Todo o código deverá estar implementado na função main.
- Não são permitidas variáveis globais.

2) Programa: cubo.cpp (2,0 pontos)

Sabe-se que um número da forma n^3 é igual à soma de n **ímpares consecutivos**. Exemplos:

- a) $1^3 = 1$
- b) $2^3 = 3 + 5$
- c) $3^3 = 7 + 9 + 11$
- d) $4^3 = 13 + 15 + 17 + 19$

Crie um programa em C que lê o valor de n e imprime os n **ímpares consecutivos** que cuja soma é igual a n^3 .

Restrição:

- a) $n \geq 1$. Se $n < 1$, então solicite novamente o valor até o usuário digitar corretamente.
- b) Todo o código deverá estar implementado na função main.
- c) Não são permitidas variáveis globais.

3) Programa: turma.cpp (3,0 pontos)

Um professor ministra aulas de Programação para duas turmas (turma A e turma B). Por um erro no sistema acadêmico, alguns alunos que se inscreveram na turma A, também foram inscritos na turma B. O professor precisa de um programa em C que leia as matrículas de todos os alunos da turma A, depois leia as matrículas de todos os alunos da turma B e, ao final, imprima:

- As matrículas dos alunos que estão inscritos na turma A e na turma B.
- As matrículas dos alunos que estão inscritos somente na turma A.
- As matrículas dos alunos que estão inscritos somente na turma B.

Restrições:

- a) Matrícula é um número inteiro positivo de 6 dígitos (zeros à esquerda não contam). Se a matrícula digitada for inválida o programa deve emitir uma mensagem de erro e solicitá-la novamente.
- b) A turma A tem n alunos, onde $n > 0$.
- c) A turma B tem k alunos, onde $k > 0$.
- d) n pode ser igual ou diferente de k .
- e) Todo o código deverá estar implementado na função main.
- f) Não são permitidas variáveis globais.

4) Programa: sequencia.cpp (3,0 pontos)

Crie um programa C para ler uma sequência **S** de **k** números inteiros, onde **k** é fornecido pelo usuário ($S = \{n_1, n_2, n_3, n_4, \dots, n_k\}$). Em seguida, o programa deve verificar se existem duas ou mais subsequências idênticas de tamanho **t** em **S** (onde **t** também é fornecido pelo usuário). Caso existam, o programa deve imprimir a sequência e a quantidade de vezes em que ela aparece. Caso contrário, o programa deve imprimir “Não existe”.

Exemplos:

k = 6
S = { 1, 2, 3, 1, 2, 3 }
t = 3
1 2 3 - 2x

k = 5
S = { 1, 2, 1, 2, 1 }
t = 2
1 2 - 2x
2 1 - 2x

k = 7
S = { 1, 2, 3, 1, 2, 3, 1 }
t = 3
1 2 3 - 2x
2 3 1 - 2x

k = 4
S = { 1, 1, 1, 1 }
t = 2
1 1 - 3x

k = 6
S = { 5, 8, 11, 13, 4, 7 }
t = 3
Não existe

Restrições:

- a) $k \geq 3$. Se $k < 3$, então solicite novamente o valor até o usuário digitar corretamente.
- b) $2 \leq t < k$. Se $t < 2$ ou $t \geq k$, então solicite novamente o valor até o usuário digitar corretamente.
- c) Todo o código deverá estar implementado na função main.
- d) Não são permitidas variáveis globais.