



1ª AVALIAÇÃO

Data: 10/11/2021

Horário: 20:20 às 23:00

Instruções para a avaliação

- Leia toda a avaliação com atenção. A interpretação das questões faz parte da avaliação.
- O arquivo do código fonte deve ser nomeado na forma: NomeDoAlunoExercicioX.c, sendo X o número da questão.
- Ao final da avaliação, somente um arquivo compactado contendo todos os códigos fonte (.c) usados para resolver cada questão deverá ser enviado para a tarefa **PRIMEIRA AVALIAÇÃO**, disponível no Moodle.
- É de responsabilidade do aluno a gravação correta dos arquivos da avaliação e o envio via Moodle.
- O tempo de avaliação é das 20h20min às 23h00min.
- A avaliação deve ser resolvida de forma **individual**.
- As respostas das questões serão submetidas a um software detector de similaridade para verificar a autenticidade do código desenvolvido pelo aluno.
- Boa Sorte!

ATENÇÃO:

- Essa avaliação tem como objetivo principal avaliar os conteúdos relacionados com a estruturas sequencial, de decisão e de repetição. Dessa forma, **os problemas deverão ser resolvidos usando apenas essas estruturas**.

- Em todas as questões comentar o código-fonte para explicar a lógica de programação.

- Além da lógica de programação, também serão avaliados: a sintaxe das estruturas, a saída esperada na solução do problema conforme o exemplo apresentado, a declaração e a inicialização das variáveis, a indentação e a legibilidade do código.

Questão 1 (4,0 pontos) – Essa questão leva em consideração os divisores de um número. Dessa forma, faça um programa que resolva os seguintes problemas:

a) Ler números e quando informado um valor 0 ou negativo parar a leitura. O valor 0 ou negativo não deve ser processado.

b) Para cada número lido, indicar se o mesmo é um número primo ou não. Sendo que para os números não primos, apresentar a quantidade de divisores que ele possui. Exemplo:

Informado	Apresentado
10	10 não é primo, possui 4 divisores
6	6 não é primo, possui 4 divisores
2	2 é primo
1	1 não é primo, possui 1 divisor

c) Verificar se o número informado pelo usuário é perfeito. Um número é perfeito quando a soma de todos os seus divisores (excluindo ele mesmo) é igual ao próprio número. Exemplo:

28 é perfeito porque $1 + 2 + 4 + 7 + 14$ é igual a 28.

Questão 2 (3,0 pontos) – Fazer um programa que receba a quantidade de caracteres (tamanho) do seu primeiro nome e use essa informação para a estrutura de repetição que lerá o nome caractere a caractere. Para cada caractere lido:

a) Verificar se o caractere é um número par ou ímpar (verificar pelo código correspondente da tabela ASCII, disponível em <https://www.ricardoarrigoni.com/tabela-ascii-completa/>). Exemplo: Quantidade de caracteres do nome Ana: 3. Usar uma estrutura de repetição (sabe-se o tamanho do nome) para ler os caracteres do nome Ana:

A – 65 na tabela ASCII é ímpar

n – 110 na tabela ASCII é par

a – 97 na tabela ASCII é ímpar

b) Contar os caracteres que se referem a vogais, para apresentar ao final o percentual de vogais informadas. Exemplo: Ana. 66,6% dos caracteres digitados são vogais.

Questão 3 (3,0 pontos) – Faça um programa que peça para o usuário informar a Idade de um aluno que será matriculado em uma nova escola. Em seguida faça:

a) Imprima a etapa de educação em que ele se insere de acordo com sua idade:

- . Idade entre 6 e 10 anos – Educação Infantil
- . Idade entre 11 e 14 anos – Ensino Fundamental
- . Idade entre 15 e 17 anos – Ensino Médio
- . Idade > 18 – Graduação

b) Utilize a idade (n) informada pelo usuário para imprimir n linhas de forma incremental, formando um triangulo composto por: n * a linha correspondente. Implementar a repetição de programa.

Exemplo para idade igual a 10.

```
10
20 20
30 30 30
40 40 40 40
50 50 50 50 50
60 60 60 60 60 60
70 70 70 70 70 70 70
80 80 80 80 80 80 80 80
90 90 90 90 90 90 90 90 90
100 100 100 100 100 100 100 100 100 100
```

Deseja continuar no programa (S/N):