

Segunda Chamada — Algor. e Prog. (10 pontos)

Prof. Me. Fernando Esquírio

Regras:

- A atividade ficará aberta no Blackboard até a data limite de entrega;
- A atividade deverá ser entregue exclusivamente pelo Blackboard;
- Prazo de entrega: 11/11/2020 até as 23:59h
- Os alunos que não tiverem acesso ao Blackboard deverão entrar em contato com o professor;
- A submissão da atividade deve ser feita em um único arquivo zip. Este arquivo deve ter o nome do aluno, ex: Joao da Silva.zip. Este zip deve conter apenas os arquivos fonte implementados para a solução dos exercícios muito bem identificados;
- Caso seu código não compile ou apresente erros de execução (crash, loop infinito, etc.), a nota daquele exercício automaticamente é zerada;
- Códigos entregues em qualquer outra linguagem que não seja C não serão corrigidos e serão zerados. Exemplos de algumas linguagens não permitidas: Java, C++, C#, Python, JavaScript, GO, Ruby, etc.
- A atividade deve ser feita de maneira individual. Caso alunos submetam códigos iguais ou muito semelhantes ou um código copiado da internet será considerado plágio. Consequentemente, a atividade inteira será zerada.
- A correção do código será feita de acordo com os critérios de correção listados abaixo. As regras da prova também serão avaliadas.
- Siga cuidadosamente as instruções de cada exercício, atentando-se para corresponder com exatidão os requisitos de cada exercício;
- Teste seu código várias vezes com várias entradas diferentes, para ter certeza de que está atendendo a todos os critérios acima;
- O seu código será avaliado não apenas no quesito funcionamento, mas também a organização, nomenclatura de variáveis, boas práticas e qualidade dos comentários.
- Estamos fazendo uma avaliação com prazo fixo de entrega, entenda que não será possível sanar dúvidas extras por e-mail ou não haverá prorrogação de prazo.
- A segunda chamada substitui apenas uma avaliação N1 perdida pelo aluno e só pode fazer a segunda chamada o aluno que perdeu alguma avaliação.

Erros comuns:

- Não use acentuação ou sinais gráficos, tanto no código quanto nos nomes dos arquivos;
- Verifique se está inicializando corretamente todas as variáveis, principalmente os acumuladores e contadores.

Critérios de avaliação dos códigos:

- 0%: Código/Funcionalidade não é válido/válida ou sem sentido ou funciona para situações bem específicas. Ou código igual ou muito semelhante a de outro aluno ou da internet.
- 50%: Código/Funcionalidade não foi completo/completa e funciona satisfatoriamente e apresenta falhas em algumas situações ou código/funcionalidade completo (a) sem comentários explicativos ou com a explicação muito simples.
- 100%: Código/funcionalidade funciona corretamente sem erros nos testes e com a explicação completa do que foi realizado.

Importante:

- Use comentários apenas para colocar os nomes dos alunos e explicar as funcionalidades implementadas.

Referência muito útil:

- Site com a documentação das bibliotecas em C: <http://www.cplusplus.com/reference/library/>

Exercício 01 (10 pontos) – Empresa do Malvado Doofenshmirtz

A Empresa do Malvado Doofenshmirtz quer fazer um código para receber um texto com até 200 caracteres do alfabeto inglês (letras maiúsculas e minúsculas, sem acentuação gráfica e caracteres especiais) e retorne a quantidade que cada letra do alfabeto aparece nesse texto. Você foi contratado para fazer o código e realizar duas funcionalidades necessárias no programa.

1. Critérios de avaliação:

(A). Código funcionando de forma simples (Diferenciando letras maiúsculas de minúsculas), recebendo o texto sem validações e verificações (Valor 5 pontos).

(B). Duas funcionalidades implementadas como descritas mais adiante.

Funcionalidades obrigatórias:

1. Não diferenciar letras maiúsculas e minúsculas, assim contabilizado em contadores distintos (Valor 2 pontos).
2. Ignorar qualquer Enter ou \n no texto digitado pelo usuário (3 pontos). Para finalizar a digitação, escolha outro critério (**Dica: use a função `fgetc(stdin)` ou a função `getche()`**).

As funcionalidades e o código serão avaliados em:

- 0%: Código/Funcionalidade não é válido/válida ou sem sentido ou funciona para situações bem específicas. Ou código igual ou muito semelhante a de outro aluno ou da internet.
- 50%: Código/Funcionalidade não foi completo/completa e funciona satisfatoriamente e apresenta falhas em algumas situações ou código/funcionalidade completo (a) sem comentários explicativos ou com a explicação muito simples.
- 100%: Código/funcionalidade funciona corretamente sem erros nos testes e com a explicação completa do que foi realizado.

Importante:

- Use comentários apenas para colocar os nomes dos alunos e explicar as funcionalidades realizadas.

Referência muito útil:

- Site com a documentação das bibliotecas em C: <http://www.cplusplus.com/reference/clibrary/>

Executáveis para vocês verem:

- Exercício 02 básico: <https://Ex02Basico.mcf1110.repl.run>
- Exercício 02 com funcionalidades obrigatórias: <https://Ex02Obrigatorio.mcf1110.repl.run>

Daqui pra baixo não faz parte da avaliação e sim um exemplo do que será avaliado

Vamos mostrar um exemplo de avaliação de funcionalidades. Não faz parte da prova. Suponha um programa que recebe um número inteiro e imprime se ele é par ou ímpar.

Exemplo de **Códigos sem funcionalidade ou com funcionalidade sem mudança significativa (valerá 0% da nota)**

Código 01	Código 02
<pre>#include <stdio.h> int main(){ int num = 0; printf("Digite um numero: "); scanf("%d", &num); if (num%2 == 0) printf("O numero %d eh par.\n", num); else printf("O numero %d eh impar.\n", num); return 0; }</pre>	<pre>#include <stdio.h> #include <stdlib.h> int main(){ int num = 0; printf("Digite um numero: "); scanf("%i", &num); fflush(stdin); if (!(num%2)) printf("O numero %d eh par.\n", num); else printf("O numero %d eh impar.\n", num); return 0; }</pre>

Exemplo de **Códigos com funcionalidade, mas não completo (sem comentários ou comentários simples) ou não funcionando corretamente (valerá 50% da nota)**

Código 03	Código 04
<pre>#include <stdio.h> int main(){ int num = 0, estado = 0; //Testa se foi digitado um numero do{ printf("Digite um numero: "); estado = scanf("%d", &num); if(estado == 0) printf("Valor invalido! \n\n"); }while(estado == 0); if (!(num%2)) printf("O numero %d eh par.\n", num); else printf("O numero %d eh impar.\n", num); return 0; }</pre>	<pre>#include <stdio.h> int main(){ int num = 0, estado = 0; char cont = 's'; //Repetir a operacao ate o usuario nao quiser mais while(cont != 'n'){ //Testa se foi digitado um numero do{ printf("Digite um numero: "); estado = scanf("%d", &num); if(estado == 0) printf("Valor invalido! \n\n"); }while(estado == 0); if (!(num%2)) printf("O numero %d eh par.\n", num); else printf("O numero %d eh impar.\n", num); printf("Deseja continuar? s-sim n-nao"); scanf(" %c", &cont); } return 0; }</pre>

Exemplo Códigos com funcionalidade e com a explicação completa da feature nova (100% da nota).

Código 05	Código 06
<pre>#include <stdio.h> int main(){ int num = 0, estado = 0; /*Testa ser o valor digitado é um número de acordo com o valor retornado pela função scanf. A função scanf retorna a quantidade de itens efetivamente lidos. Dessa forma, se o valor o código de formatação é %d, a função retorna 0 0 para qualquer valor digitado que não seja um número e diferente de 0 para um número. */ do{ printf("Digite um numero: "); estado = scanf("%d %d", &num, &num); printf("Estado: %d \n\n", estado); fflush(stdin); if(estado == 0) printf("Valor invalido! \n\n"); }while(estado == 0); if (!(num%2)) printf("O numero %d eh par.\n", num); else printf("O numero %d eh impar.\n", num); return 0; }</pre>	<pre>#include <stdio.h> int main(){ int num = 0, estado = 0; char cont = 's'; /*Repete o loop enquanto não for digitado o caractere n. Para qualquer outro caractere digitado diferente de n o laço continua, não necessariamente deve ser s. */ while(cont != 'n'){ /*Testa ser o valor digitado é um número de acordo com o valor retornado pela função scanf. A função scanf retorna a quantidade de itens efetivamente lidos. Dessa forma, se o valor o código de formatação é %d, a função retorna 0 0 para qualquer valor digitado que não seja um número e diferente de 0 para um número. */ do{ printf("Digite um numero: "); estado = scanf("%d %d", &num, &num); printf("Estado: %d \n\n", estado); fflush(stdin); if(estado == 0) printf("Valor invalido! \n\n"); }while(estado == 0); if (!(num%2)) printf("O numero %d eh par.\n", num); else printf("O numero %d eh impar.\n", num); printf("Deseja continuar? s-sim n-nao"); scanf(" %c", &cont); } return 0; }</pre>