UNIRIO – Universidade Federal do Estado do Rio de Janeiro
Bacharelado em Sistemas de Informação
Técnicas de Programação I - 02/2021
Prof. Jobson Massollar

Trabalho 3

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 06/02/2022 (domingo).
- A data de entrega não será adiada.
- Os 3 programas solicitados (arquivos com extensão CPP) deverão ser compactados em um único arquivo (ZIP ou RAR) com o nome e sobrenome do aluno.
- Não envie outros arquivos. Somente os arquivos com extensão CPP.
- 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.
- O programa que não obedecer às restrições estabelecidas receberá zero.

1) Programa: embaralha.cpp (3,0 pontos)

Crie um programa C que lê uma palavra (cadeia de caracteres) e embaralha as letras dessa palavra, ou seja, gera outra palavra com exatamente as mesmas letras e respectivas quantidades, mas em posições diferentes escolhidas aleatoriamente. Ao final o programa deverá imprimir a palavra original e a palavra embaralhada. Exemplos:

a) programacao $\rightarrow$ goarcmaarop

b) computacao $\rightarrow$ ctupamocoa

Dica: para gerar uma posição aleatória, pesquise na lista de exercícios sobre funções.

O programa deverá ter, obrigatoriamente, as seguintes funções:

- **verifica_palavra**: que deverá verificar se a cadeia de caracteres digitada pelo usuário é uma palavra válida ou não. Para ser considerada uma palavra, a cadeia de caracteres deve ter de 5 a 50 caracteres e conter apenas letras de a..z (caixa alta ou baixa).
- **embaralha**: que retorna a palavra fornecida pelo usuário embaralhada.

Restrições:

a) O programa deverá ler cadeias de caracteres e processá-las até o usuário digitar string vazia, ou seja, o programa só termina quando o usuário digitar string vazia.

b) Se a cadeia de caracteres fornecida pelo usuário não for uma palavra, o programa deverá imprimir uma mensagem de aviso "Não é uma palavra" e ignorá-la.

c) Outras funções podem ser implementadas a critério do aluno.

d) Podem ser usadas funções da biblioteca do C.

e) Não podem ser usadas variáveis globais, somente constantes globais.

2) Programa: fetuccine.cpp (4,0 pontos)

A série de Fetuccine modificada é gerada da seguinte forma: os três primeiros números da série devem ser inteiros positivos fornecidos pelo usuário. A partir daí, os termos da série são gerados pela soma e pela subtração dos três termos anteriores (iniciando com a soma e alternando essas operações). Exemplos:

c) 3 7 5 15 -13 7 21 15 -29 7 ...

d) 1 2 3 6 -7 2 11 6 -15 2 ...

Crie um programa em C para ler um valor inteiro **n** (n > 4) e imprimir de forma invertida a série de Fetuccine modificada com n termos.

O programa deverá ter, obrigatoriamente, as seguintes funções:

- **le_termos_iniciais**: que deverá retornar os três primeiros termos da série de Fetuccine modificada fornecidos pelo usuário.
- **calcula_serie**: que deverá retornar os n termos da série de Fetuccine.
- **inverte_serie**: que deverá receber a série de Fetuccine e retorná-la invertida.
- **imprime_serie**: que deverá receber a série já invertida e imprimi-la em uma única linha, separando seus elementos por um espaço em branco.

Restrições:

a) n > 4, caso contrário solicite novamente o valor até o usuário digitar corretamente.

b) Outras funções podem ser implementadas a critério do aluno.

c) Podem ser usadas funções da biblioteca do C.

d) Não podem ser usadas variáveis globais, somente constantes globais.

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

Crie um programa em C para ler um valor inteiro **n** ($n \geq 4$), ler as coordenadas de n pontos (x, y) no plano cartesiano e imprimir as coordenadas do centro (x_c, y_c) e o raio da circunferência mínima que circunscreve todos esses pontos. O centro da circunferência deverá ser, obrigatoriamente, um dos pontos fornecidos pelo usuário.

Exemplo:

- Entrada
  5
  1.0 0
  -0.5 0
  0 0
  -1.0 1.0
  0 -1.0
- Saída
  Centro: (0.0000, 0.0000)
  Raio: 1.4142

O programa deverá ter, obrigatoriamente, as seguintes funções:

- **le_pontos**: que deverá retornar os n pontos do plano cartesiano fornecidos pelo usuário.
- **distancia**: que deverá retornar a distância euclidiana entre dois pontos.
- **calcula_circunferencia**: que deverá calcular e retornar as coordenadas do centro (x_c, y_c) e o raio da circunferência mínima que circunscreve todos os pontos fornecidos pelo usuário.

Restrições:

a) $n \geq 4$, caso contrário solicite novamente o valor até o usuário digitar corretamente.

b) O programa deverá usar uma struct para representar um ponto (x, y) no plano cartesiano.

c) Valores reais deverão ser impressos com 4 casas decimais.

d) Outras funções podem ser implementadas a critério do aluno.

e) Podem ser usadas funções da biblioteca do C.

f) Não podem ser usadas variáveis globais, somente constantes globais.