

## Trabalho 2

### LEIA ATENTAMENTE AS REGRAS E OS ENUNCIADOS

<b>R E G R A S</b>
--------------------

- O trabalho deverá ser realizado individualmente.
- O trabalho deverá ser enviado para o Google classroom até o dia 16/01/2022 (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: subsequencia.cpp (3,0 pontos)

Leia um valor inteiro  $n$  ( $n \geq 2$ ), uma sequência  $L$  de  $n$  números inteiros e, por fim, um valor inteiro  $k$  ( $2 \leq k \leq n$ ). Ao final, imprima as subsequências de  $k$  números consecutivos de  $L$  com o maior e com o menor somatório e os respectivos somatórios. Se mais de uma subsequência tem o maior somatório, imprima a primeira delas. Se mais de uma subsequência tem o menor somatório, imprima a última delas.

Exemplo:

$L = [ 6, 2, -4, 9, 3, 12, 5, 1, -2, 9, 15, 1, 8, 6, 7, 12 ]$

$k = 3$

Subsequência de 3 números consecutivos de maior soma =  $[ 9, 15, 1 ]$  (soma = 25)

Subsequência de 3 números consecutivos de menor soma =  $[ 5, 1, -2 ]$  (soma = 4)

No exemplo acima note que:

- A sequência  $[6, 7, 12]$  também soma 25, mas foi impressa a primeira encontrada.
- A sequência  $[6, 2, -4]$  também soma 4, mas foi impressa a última encontrada.

Restrições:

- $n \geq 2$ , caso contrário solicite novamente o valor até o usuário digitar corretamente.
- $2 \leq k \leq n$ , caso contrário solicite novamente o valor até o usuário digitar corretamente.
- Todo o código deverá estar implementado na função main, sem o uso de funções auxiliares implementadas por você. Podem ser usadas funções da biblioteca do C.

## 2) Programa: calculadora.cpp (4,0 pontos)

Crie um programa em C para ler expressões matemáticas e imprimir seus resultados até o usuário digitar uma string vazia. Para cada expressão lida o programa deverá calcular e imprimir seu resultado ou a mensagem de erro "Expressão inválida", caso a expressão esteja sintaticamente errada. A expressão tem as seguintes características:

- i. É composta apenas de números inteiros de 1 dígito.
- ii. Possui 5 operadores binários: + (soma), - (subtração), \* (multiplicação), / (divisão real) e ^ (exponenciação).
- iii. Não existe o operador - unário.
- iv. Podem existir ou não espaços em branco separando operandos e operadores.
- v. Não há precedência de operadores, ou seja, as operações devem ser calculadas à medida em que são encontradas na expressão.

Exemplos:

- $2 + 3 * 6 \rightarrow 30.000000$
- $2 * 4 ^ 2 \rightarrow 64.000000$
- $5 - 7 * 5 / 3 \rightarrow -3.333333$
- $7 / 2 * 3 \rightarrow 10.500000$
- $4 * - 8 \rightarrow \text{Expressão inválida}$
- $3 \ 2 * 7 \rightarrow \text{Expressão inválida}$
- $9 * 7 , 8 \rightarrow \text{Expressão inválida}$
- $- 7 * 4 \rightarrow \text{Expressão inválida}$

Atenção: apesar de ser composta de números inteiros, o resultado da expressão pode ser um número real.

Restrições:

- a) A expressão deve ser lida como uma cadeia de caracteres.
- b) Assuma que a expressão tem, no máximo, 100 caracteres.
- c) Imprima o resultado com 6 casas decimais.
- d) Todo o código deverá estar implementado na função main, sem o uso de funções auxiliares implementadas por você. Podem ser usadas funções da biblioteca do C.

3) Programa: caca\_numero.cpp (3,0 pontos)

Leia uma matriz **m** de números inteiros de **N x N** ( $N \geq 5$ ). Em seguida, leia uma sequência **s** com **k** números inteiros ( $2 \leq k \leq N$ ). Por fim, mostre todas as posições (**x**, **y**) de **s** em **m** e o sentido (direita para esquerda, esquerda para direita, cima para baixo ou baixo para cima). Os valores **x** e **y** deverão ser os índices da **linha** e **coluna** da matriz **m** onde **s** inicia. Caso **s** não exista em **m** imprima a mensagem "Sequência não encontrada". Exemplo:

N = 5

m = 

4	9	12	7	23
2	23	7	23	2
23	9	14	2	6
7	5	2	11	19
15	14	7	23	2

k = 3

s = [ 7, 23, 2 ]

(1, 2) sentido: esquerda para direita

(4, 2) sentido: esquerda para direita

(1, 2) sentido: direita para esquerda

(0, 3) sentido: cima para baixo

(3, 0) sentido: baixo para cima

Restrições:

- a)  $2 \leq k \leq N$ , caso contrário solicite novamente o valor até o usuário digitar corretamente.
- b) Todo o código deverá estar implementado na função main, sem o uso de funções auxiliares implementadas por você. Podem ser usadas funções da biblioteca do C.
- c)  $N \geq 5$ . A constante **N** e a matriz **m** deverão ser definidas conforme trecho de código a seguir:

```
const int N = 5;

int main() {
    int m[N][N];

    // Restante do código aqui!
}
```

**Importante:** durante a correção da questão o valor da constante **N** será modificado para outros valores como 8, 10, 12, etc. Assim, seu código deverá funcionar para quaisquer valores de **N** onde  $N \geq 5$ .