

# UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

## INSTITUTO METRÓPOLE DIGITAL

Linguagem de Programação I • DIM0120

◁ Exercícios de Programação #1 ▷

1 de agosto de 2018

1. Escreva um programa em C++ chamado `negativo5.cpp` que lê 5 valores inteiros, um de cada vez, conta quantos destes valores são negativos e imprime esta informação.
2. Escreva um programa em C++ chamado `intervalos.cpp` que lê um número não conhecido de valores, um de cada vez, e conta quantos deles estão em cada um dos intervalos  $[0, 25)$ ,  $[25, 50)$ ,  $[50, 75)$  e  $[75, 100]$ .

Para encerrar a entrada de dados o usuário deve pressionar <Ctrl+d>. Para ler valores do terminal até o usuário digitar <Ctrl+d> você pode utilizar o seguinte trecho de código:

```
int x;
...
cout << "Entre com valores inteiros (Ctrl+d p/ encerrar): " << endl;
while( cin >> x ) {
    // Realização da contagem de ocorrências nos intervalos
    ...
}
// Exibir contagem para os intervalos solicitados.
```

Após encerrada a entrada de dados, o programa deve imprimir a porcentagem de números para cada um dos quatro intervalos indicados.

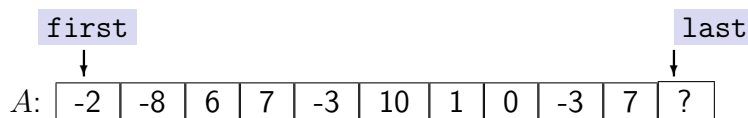
3. Escreva um programa em C++ chamado `soma_pares.cpp` que lê um número não determinado de pares " $m\ n$ " (sem as aspas), todos inteiros e positivos, um par de cada vez, calcula e escreve a soma dos  $n$  primeiros inteiros consecutivos à partir de  $m$  (inclusive). Para encerrar a entrada de dados você deve utilizar <Ctrl+d>. Por exemplo, se uma entrada for "3 5" o programa deve calcular a soma dos 5 primeiros inteiros a partir de 3 (inclusive), ou seja,  $3 + 4 + 5 + 6 + 7 = 25$  e imprimir como resultado 25. Assuma que a entrada sempre é válida.
4. Implemente um programa em C++ chamado `fib_b4_n.cpp` que recebe um valor inteiro positivo  $n$  e imprime os termos da série de Fibonacci **inferiores** a  $n$ .

A sequência de Fibonacci define-se como tendo os dois primeiros termos iguais a 1 e cada termo seguinte é a soma dos dois termos imediatamente anteriores. Desta forma se

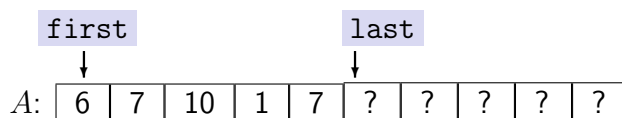
fosse fornecido ao programa uma entrada  $n = 15$  o mesmo deveria produzir a seguinte sequência de termos da série: 1 1 2 3 5 8 13.

5. Escreva um programa em C++ chamado `menor_elemento.cpp` que lê 20 números reais, os armazena em um arranjo unidimensional (vetor) `Vet` e o imprime na tela. A seguir, o programa deve encontrar o menor elemento e a sua posição no vetor `Vet` e escrever na saída padrão qual é o menor elemento e que posição ele ocupa no vetor.
6. Escreva um programa em C++ chamado `troca_interna.cpp` que lê 20 inteiros, os armazena em um arranjo unidimensional (vetor) `A` e o imprime na tela. A seguir, o programa deve trocar o conteúdo do último elemento de `A` com o 1º, do penúltimo com o 2º, do antepenúltimo com o 3º e assim por diante até que todos os elementos tenham sido trocados de lugar *apenas uma vez*. Por fim, o programa deve imprimir o vetor modificado.
7. Escreva um programa em C++ chamado `troca_seguintes_vet.cpp` que lê 20 inteiros, os armazena em um arranjo unidimensional (vetor) `B` e o imprime na tela. A seguir, o programa deve trocar o conteúdo dos elementos de `B` de ordem ímpar com os de ordem par imediatamente seguintes e imprimir o vetor modificado.
8. Escreva uma função em C++ `filter` que “filtra” os elementos no intervalo `[first; last)` definido sobre um vetor de inteiros por meio de ponteiros, retirando todos os **valores nulos e negativos** e preservando a ordem relativa dos elementos filtrados. A função deve retornar um ponteiro para a posição após o último elemento que permaneceu no vetor depois de realizada a operação de filtragem.

Considere o exemplo abaixo com apenas 10 elementos no intervalo:



depois de filtrado o intervalo fica com tamanho “lógico” = 5.



~ FIM ~