

2020_1 - PROGRAMAÇÃO E DESENVOLVIMENTO DE SOFTWARE II - METATURMA

PAINEL > **MINHAS TURMAS** > **2020_1 - PROGRAMAÇÃO E DESENVOLVIMENTO DE SOFTWARE II - METATURMA**
> **FUNDAMENTOS - DATA LIMITE: 31 DE AGOSTO. TOTAL DE 19 PONTOS.** > **AULA 05 - VPL 01 - LISTA PROBLEMÁTICA (4 PTS)**

[Descrição](#)[Enviar](#)[Editar](#)[Visualizar envios](#)

Aula 05 - VPL 01 - Lista Problemática (4 pts)

Data de entrega: segunda, 31 Ago 2020, 23:59

Arquivos requeridos: main.cpp, List.hpp, List.cpp ([Baixar](#))

Tipo de trabalho: Trabalho individual

Descrição:

Sua tarefa é modificar a função **main** para que ela execute as tarefas abaixo listadas, e corrigir a implementação do arquivo **List.cpp**.

O programa apresentado nesse exercício contém múltiplos erros intencionais para que seja possível reproduzir problemas muito comuns como **segmentation fault**, **memory leak**, e **null pointer**. Faz parte do exercício a interpretação e compreensão mais detalhada do código.

Tarefas:

- 1) Receba dois valores como entrada, **N** e **K**.
- 2) Insira na lista os valores de **[0,N]**. Sua lista deve sempre inserir os números na primeira posição, de forma com que ela seja decrescente quando exibida sequencialmente. **Atenção:** o intervalo **[0,N]** é fechado!
- 3) Imprima os valores contidos na lista.
- 4) Imprima o tamanho da lista.
- 5) Remova os **K** primeiros valores.
- 6) Imprima os valores contidos na lista.
- 7) Imprima o tamanho da lista.
- 8) Remova os valores **pares** existentes na lista.
- 9) Imprima os valores contidos na lista.
- 10) Imprima o tamanho da lista.

Atenção:

- 1) Se a lista estiver vazia o programa deve imprimir uma linha com o caractere "-" (veja o terceiro exemplo) para a representar.
- 2) Nenhuma função não especificada no TAD é necessária. O uso de qualquer função extra será penalizado.
- 3) O seu programa será avaliado automaticamente em relação a saída, e manualmente para o erro de **memory leak** utilizando o **Valgrind**. Alunos que não resolverem problemas relacionados a **memory leak** serão penalizados no valor da nota final. Utilize o botão 'Depurar' para acessar o GDB e visualizar a saída do Valgrind.

Exemplos de execução:

Entrada	Saída
5 3	5 4 3 2 1 0 6 2 1 0 3 1 1
3 0	3 2 1 0 4 3 2 1 0 4 3 1 2
0 0	0 1 0 1 - 0

Arquivos requeridos

main.cpp

```
1 #include <iostream>
2 #include "List.hpp"
3
4 int main(){
5     List *L = new List();
6     delete L;
7     return 0;
8 }
```

List.hpp

```
1 #ifndef LIST_H
2 #define LIST_H
3
4 #include <iostream>
5 #include <cstdlib>
6
7 struct Node{
8     int _value;
9     Node *_next;
10 };
11
12 struct List{
13     Node *_head;
14     int _size;
15
16     List();
17     void insert(int value);
18     void remove(int value);
19     void print();
20     void clearList();
21     int removeFirst();
22     int indexOf(int value);
23 };
24
25 #endif
```

List.cpp

```
1  #include "List.hpp"
2
3  List::List(){
4      _size = 0;
5      _head = NULL;
6  }
7
8  void List::insert(int value){
9      Node* node;
10     node->value = value;
11     node->next = _head;
12     _head = node;
13 }
14
15 void List::clearList() {
16     _head = NULL;
17 }
18
19 void List::print(){
20     Node* head = _head;
21     if (_size > 0){
22         while(head){
23             std::cout << head->value << " ";
24             head = head->next;
25         }
26         std::cout<<std::endl;
27     }
28 }
```

[VPL](#)

◀ How to Think Like a Computer
Scientist using C++ (Capítulos 10, 12 e
13)

Seguir para...

(Tutorial) Fazendo uso do Valgrind.
Ajuda neste VPL ►

