## 

**Nome**: Lucas Silva de Jesus **RA**: 202104644604 **Curso**: Sistema de Informação

**Nome**:TIAGO SALLES DA SILVA **RA**: 202104192479 **Curso**: ENGENHARIA DA COMPUTAÇÃO

## **Sumário**

[**Tabela da Média do Algoritmos nos Arquivos.**](#_wsc7y3apihiu) **2**

[**BubbleSort**](#_86fvwgdycueg) 3

[desordenado5k](#_wr7117w9a7zt) 3

[desordenado25k](#_va4xcfh2r8v2) 5

[desordenado125k](#_8vfscev14dzb) 8

[ordenado5k](#_6dey0019iqpy) 10

[ordenado25k](#_2grlbl8f68ak) 13

[ordenado125k](#_lqaawihk36o4) 15

[ordenadodesc5k](#_2904khmju3j9) 18

[ordenadodesc25k](#_86kzlxhqmd01) 20

[**InsertionSort**](#_fa3tsek28w3q) 23

[desordenado5k](#_xhodnmq02yve) 23

[desordenado25k](#_mdiz0byo1ujf) 25

[desordenado125k](#_g8hahd12u4fw) 28

[ordenado5k](#_ujpwl3ojomcm) 30

[ordenado25k](#_8ddefjf3ui5w) 33

[ordenado125k](#_8rzds4mcgk9m) 35

[ordenadodesc5k](#_8i6p0njpno2p) 38

[ordenadodesc25k](#_j47vrvosnl05) 40

[**SelectionSort**](#_qu6un96gb9lq) 43

[desordenado5k](#_xrqwafykmv6y) 43

[desordenado25k](#_2ekq0xpyxabz) 45

[desordenado125k](#_yxbzaoh3v8lk) 48

[ordenado5k](#_s9n05zlnq2kf) 50

[ordenado25k](#_5c1bnf1zk3e) 53

[ordenado125k](#_jwu6ie507xb) 55

[ordenadodesc5k](#_e831txyef4zd) 58

[ordenadodesc25k](#_5o4dkg4lr799) 60

[**MergeSort**](#_7wynxjcy4vfn) 63

[desordenado5k](#_gyxy451jnn9o) 63

[desordenado25k](#_h786vf6dt8dr) 65

[desordenado125k](#_tengvepsrgh8) 68

[ordenado5k](#_3k7dt4amp49g) 70

[ordenado25k](#_fzavomfx0a2g) 73

[ordenado125k](#_wa2k1m5it9ar) 76

[ordenadodesc5k](#_xzpiyja382i9) 78

[ordenadodesc25k](#_mz0a6370ovsc) 81

[**HeapSort**](#_96acmknmp3g7) 83

[desordenado5k](#_rayz3lb20qqi) 83

[desordenado25k](#_1zbsbjfsagam) 86

[desordenado125k](#_ie1k1zhvtn11) 88

[ordenado5k](#_q8ztbjadayl6) 91

[ordenado25k](#_uqh1x8sby5i) 93

[ordenado125k](#_c61dvif5vk10) 96

[ordenadodesc5k](#_7lka5k1qz1cb) 98

[ordenadodesc25k](#_rpttvmv9336) 101

[**QuickSort**](#_yoqeqk27jbvb) 103

[desordenado5k](#_3gr64d49ptnh) 103

[desordenado25k](#_88dnle2n6hcg) 106

[desordenado125k](#_2wnhunav0q6v) 108

[ordenado5k](#_dajt2zt9mg87) 111

[ordenado25k](#_vgf533qcr1bs) 113

[ordenado125k](#_uh5i4di7761z) 115

[ordenadodesc5k](#_14c8zmdnvjve) 116

[ordenadodesc25k](#_fl85rm9rpfq0) 118

## 

## **Tabela da Média do Algoritmos nos Arquivos.**

|  | **Bubble Sort** | **Insertion Sort** | **Selection Sort** | **Merge Sort** | **Heap Sort** | **Quick Sort** |
| --- | --- | --- | --- | --- | --- | --- |
| **desordenado5k.txt** | 48000 µs | 15501µs | 31000µs | 970µs | 1000µs | 1000µs |
| **desordenado25k.txt** | 16211992µs | 370000µs | 782001µs | 4000µs | 5995µs | 4000µs |
| **desordenado125k.txt** | 42400000µs | 9000000µs | 18625042µs | 23000µs | 36001µs | 18000µs |
| **ordenado5k.txt** | 337966µs | 0µs | 30000µs | 1000µs | 1000µs | 66997µs |
| **ordenado25k.txt** | 828000µs | 0µs | 759998µs | 2000µs | 4000µs | 1666964µs |
| **ordenado125k.txt** | 20160000µs | 999µs | 18764968µs | 13000µs | 29000µs | - |
| **ordenadodesc5k.txt** | 38000µs | 29000µs | 29001µs | 970µs | 1000µs | 49999µs |
| **ordenadodesc25k.txt** | 954000µs | 751976µs | 741970µs | 1000µs | 4500µs | 1265001µs |
| **ordenadodesc125k.txt** | - | - | - | - | - | - |

Observação: O ordenado125k não executou em nenhuma das máquinas dos dois integrantes por isso, não há evidências e ordenadodesc125k não tem arquivo o arquivo de texto.

### 

### **Análise dos Algoritmos**

Analisando a partida da tabela anterior se dá para ter uma noção da diferença de performance de cada algoritmo de ordenação, irei começar discursando do pior (mais demorado) para o melhor (menos demorado).

O Bubble Sort analisando dentre todos os testes efetuados dá para reparar que é algoritmo menos performático dentres os apresentados, isso se dá porque a maneira em como ele faz sua ordenação é extremamente gastona, onde na PIOR das hipóteses terá de percorrer *n* vezes a estrutura sendo *n* o número de elementos contidos dentro da estrutura utilizada.

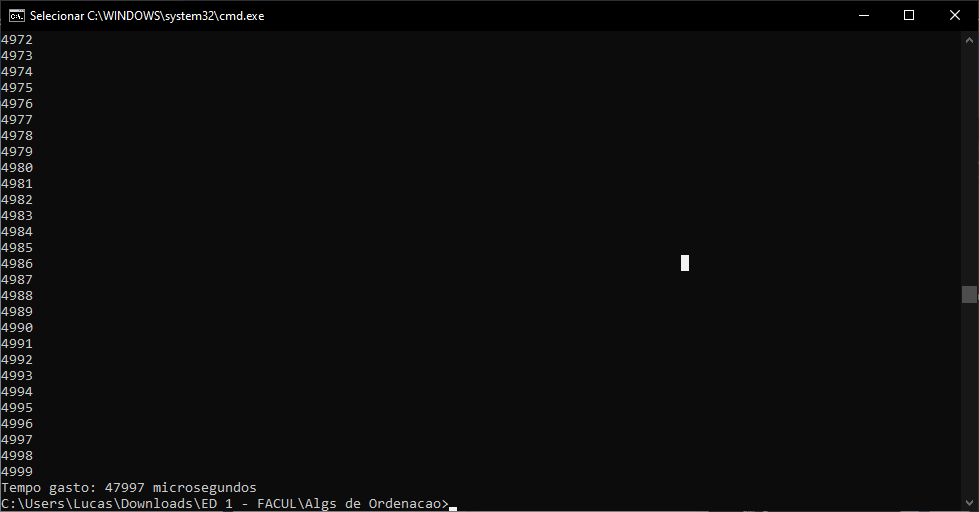
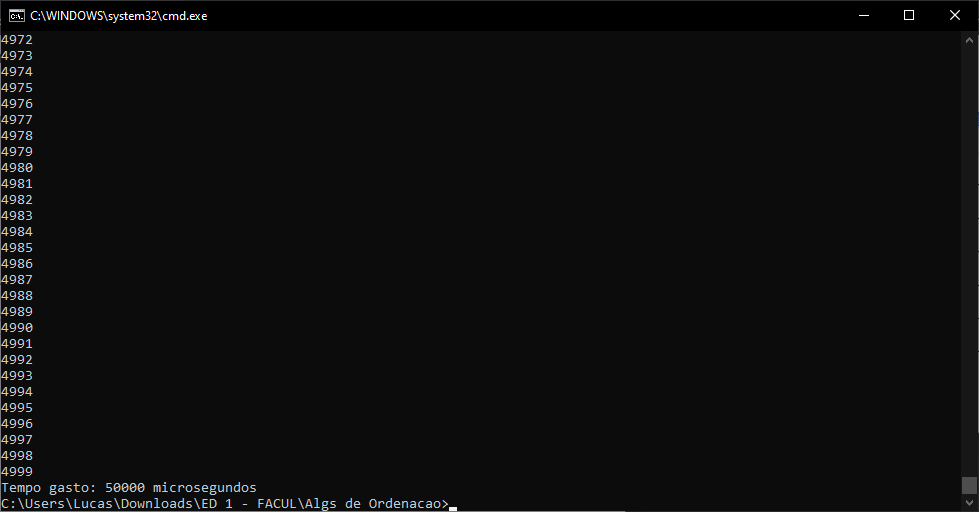
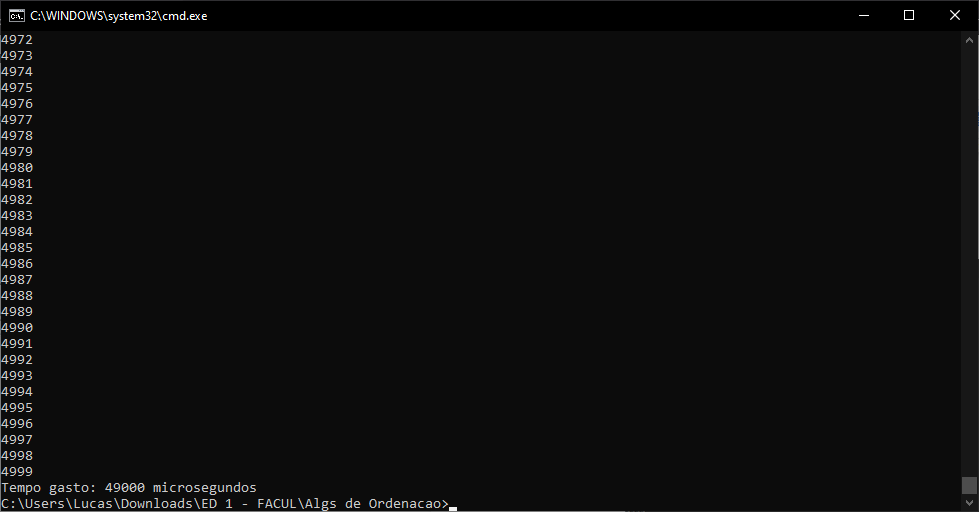
Já o Insertion, Selection e Heap sort são algoritmos com estratégias de ordenação diferentes mas com resultados não tão diferentes entre si. Sendo TODOS melhores que o Bubble Sort mas piores que o Merge e Quick Sort.

Partindo para os finais os dois Algoritmos mais performáticos são o Quick e Merge sort sendo nesta sequência de melhores desempenhos. Por terem estratégias mais inteligentes para lidar com as estruturas que serão ordenadas, conseguem evitar gastos desnecessários. Um exemplo é que o quicksort na PIOR das hipóteses irá percorrer no máximo, *n/2* ou a metade da estrutura para fazer sua ordenação.

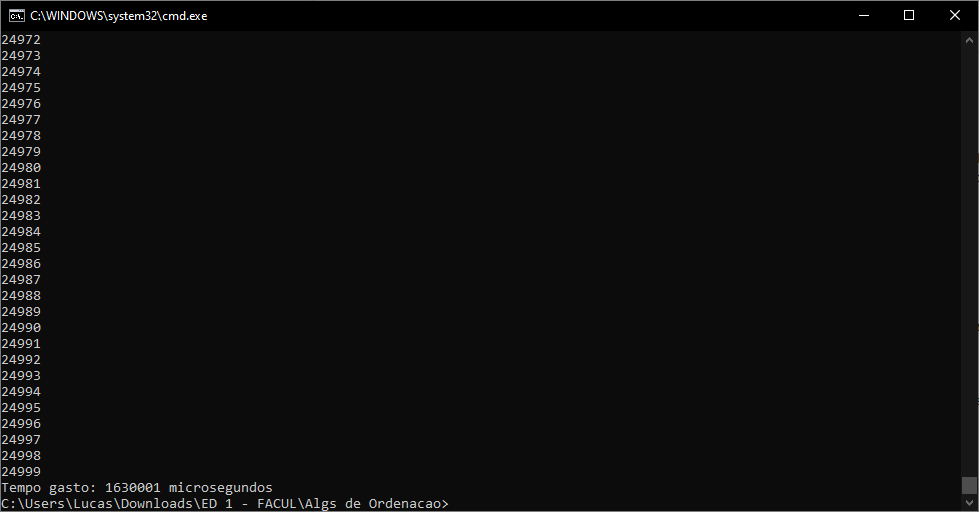
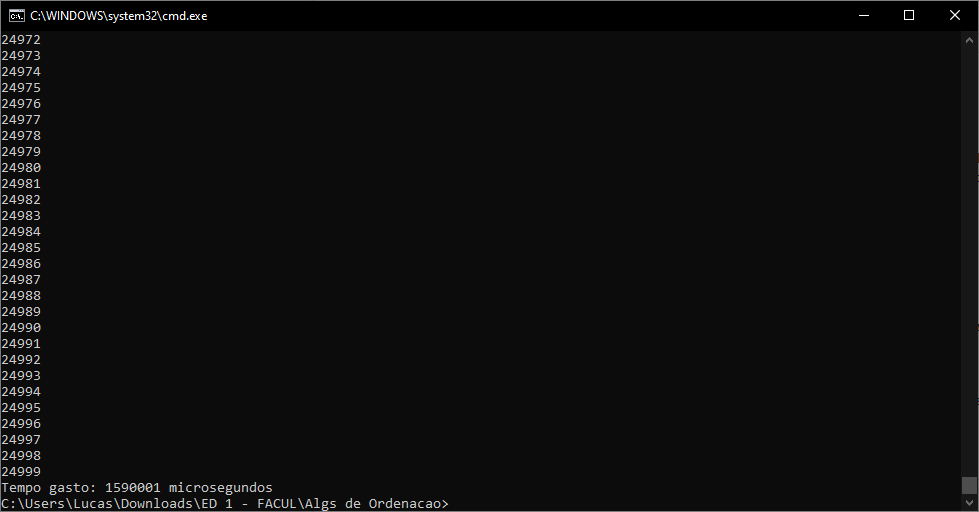
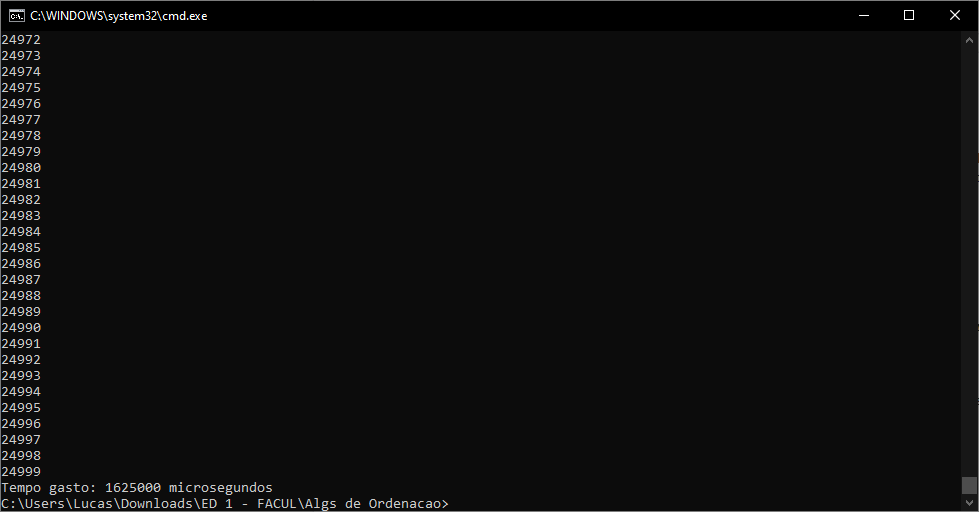
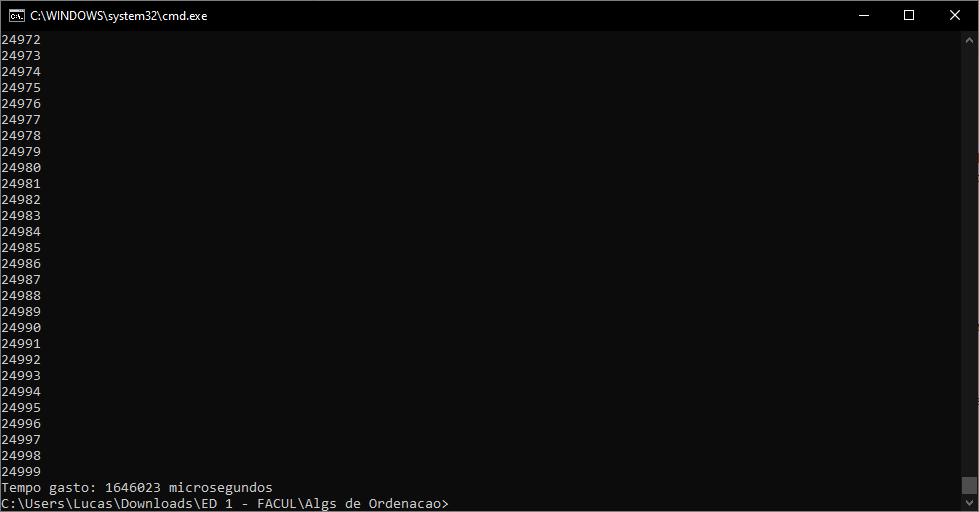
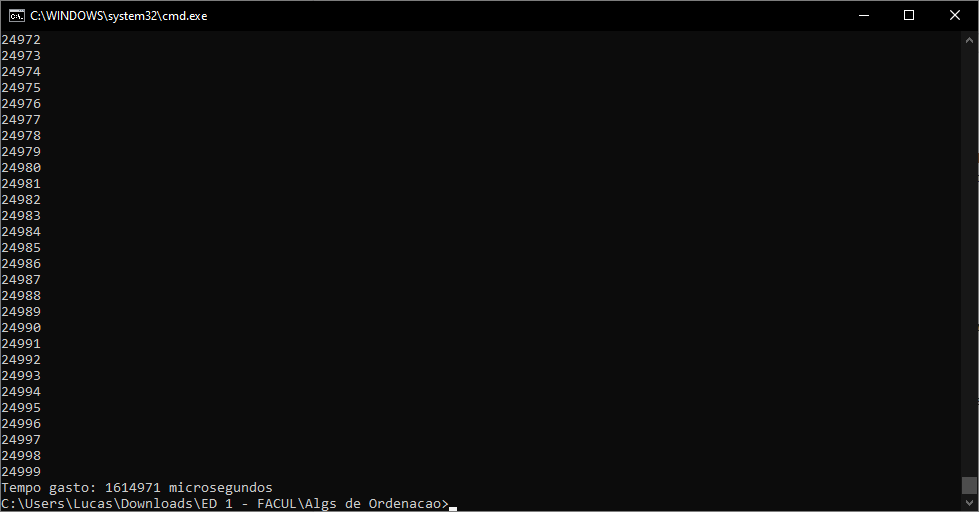
Só para registrar enquanto executava os Arquivos com 125k de itens, deixava ele executando por alguns bons segundos, a ponto de ir na cozinha e voltar. Já com o quicksort, demorava alguns segundos também mas bem menos tempo, a ponto de apenas pegar o celular e abrir a rede social tik tok, ao olhar de volta já havia sido finalizado.

### ***BubbleSort***

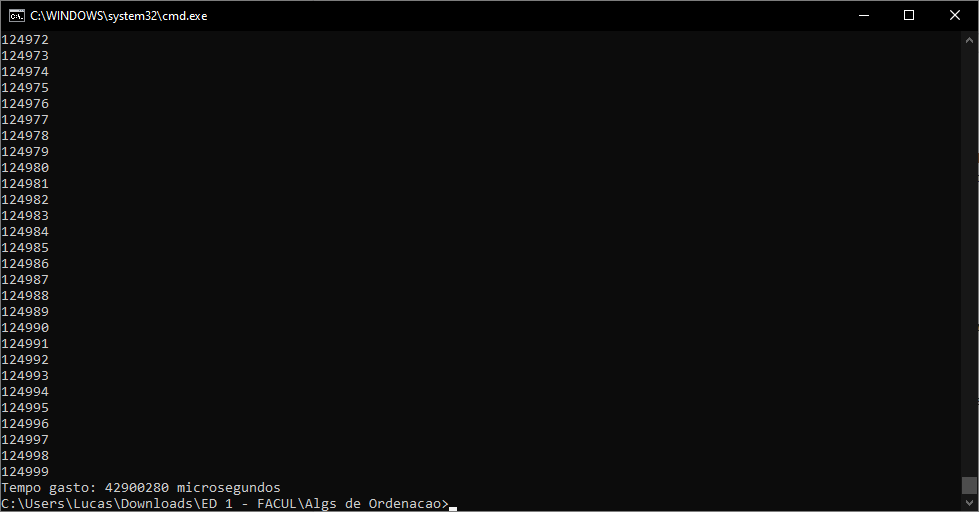
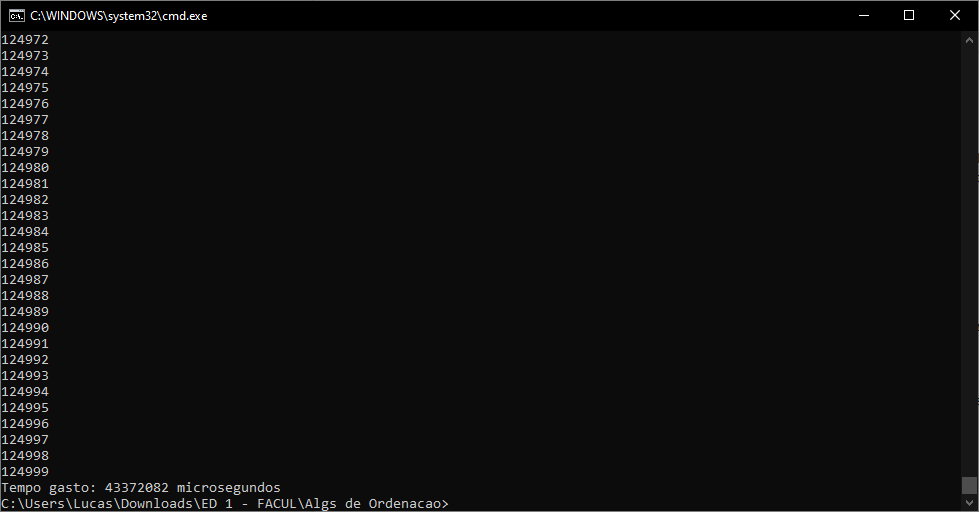
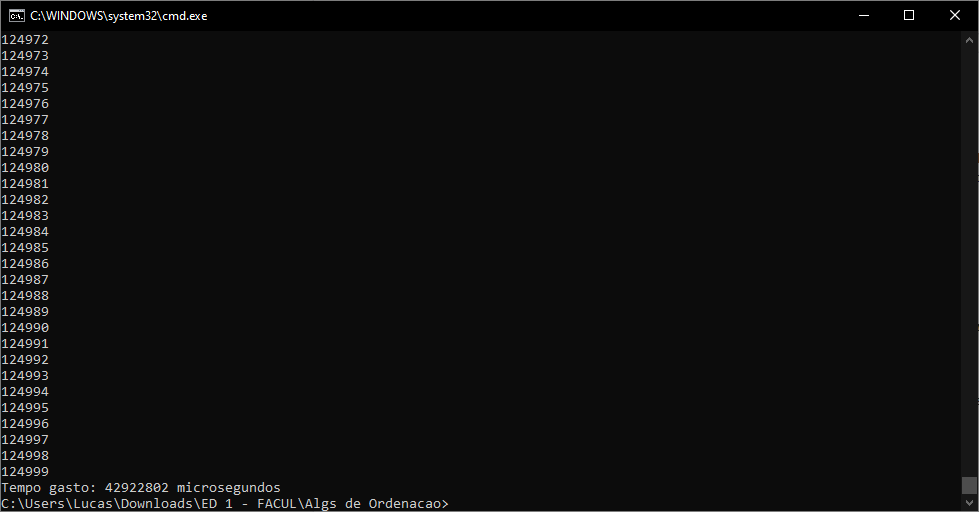
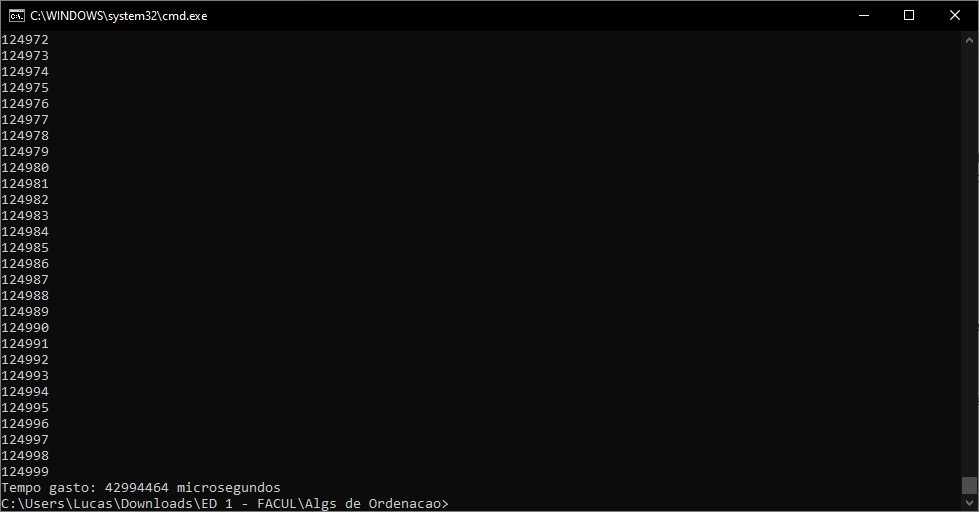
#### ***desordenado5k***

1. 
2. 
3. 
4. 
5. 

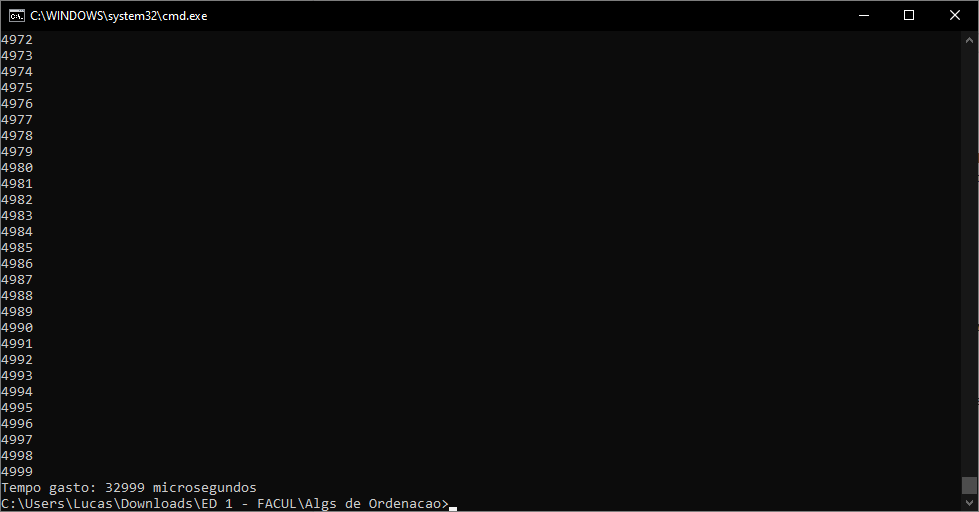
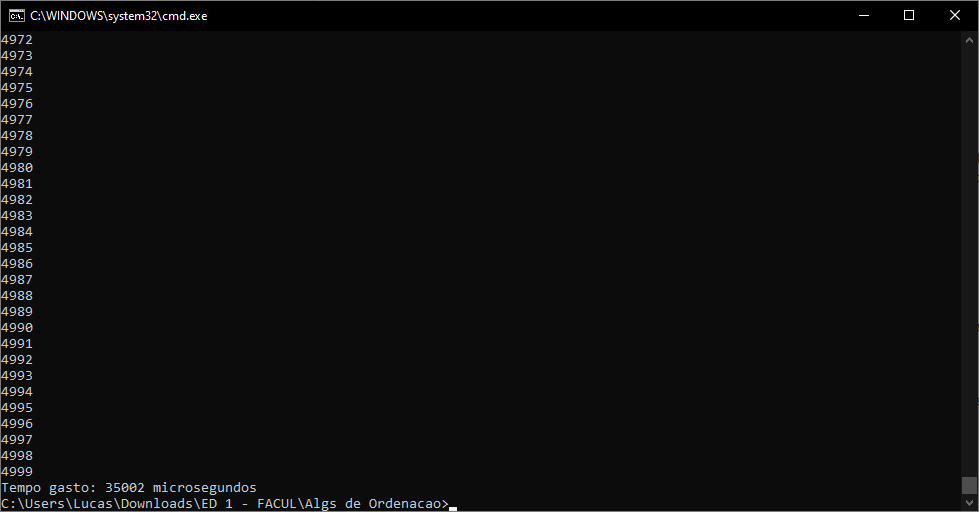
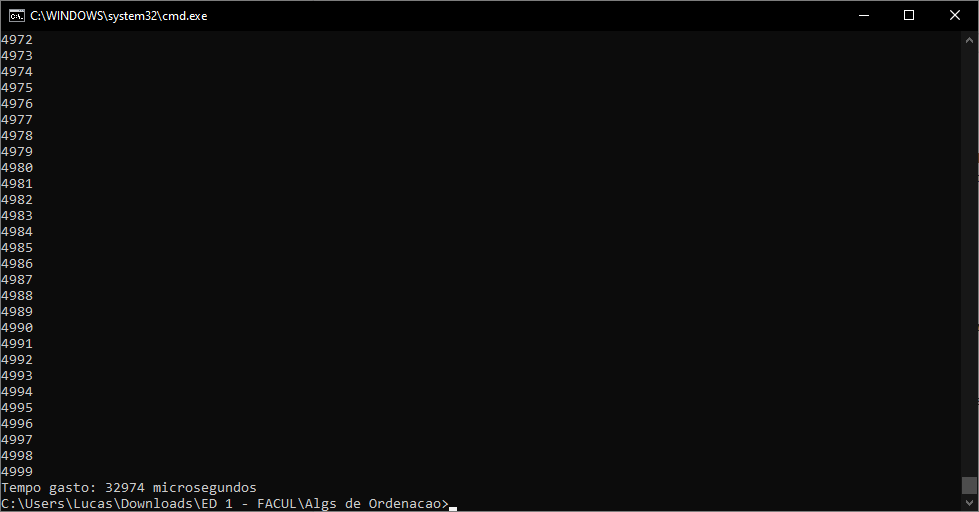
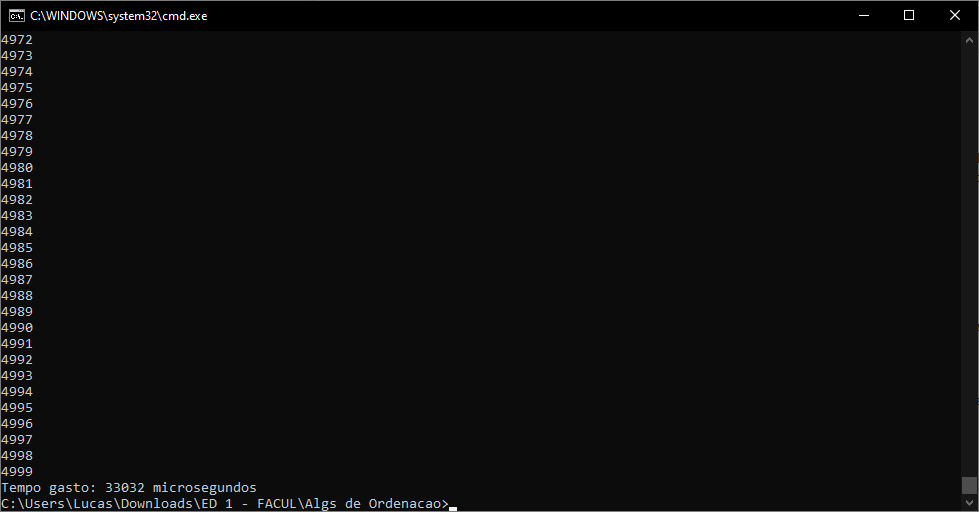
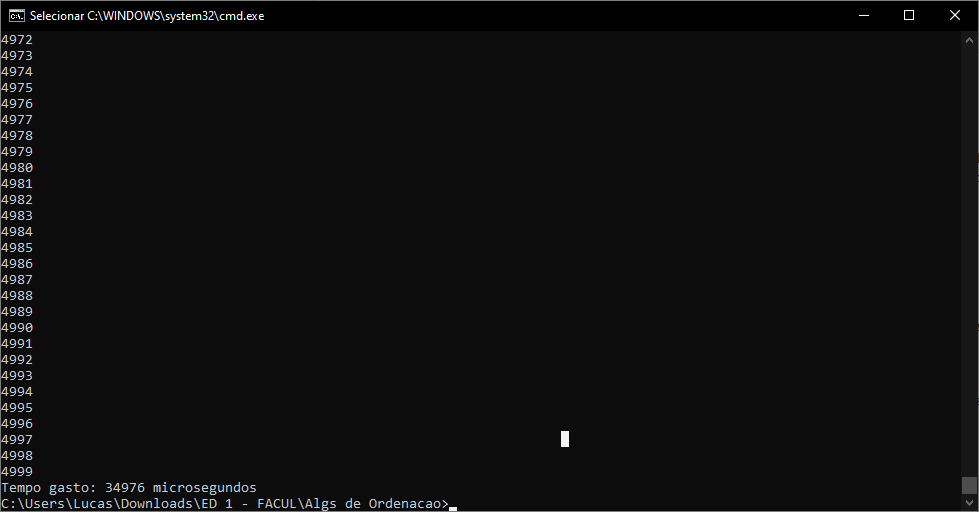
#### **desordenado25k**

1. 
2. 
3. 
4. 
5. 

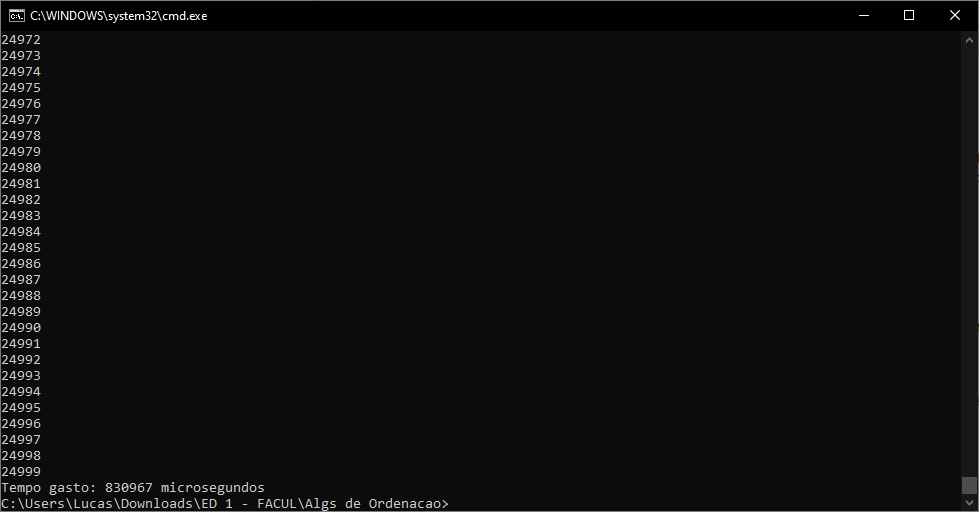
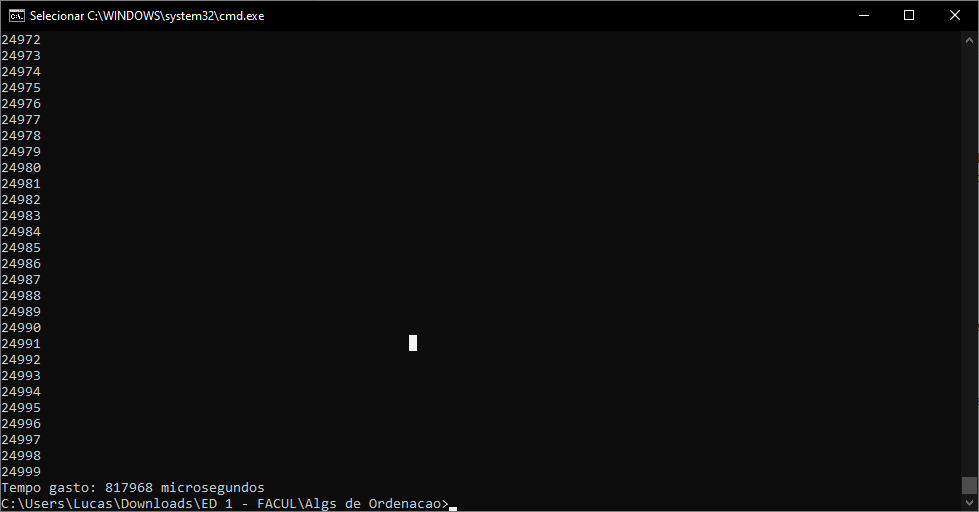
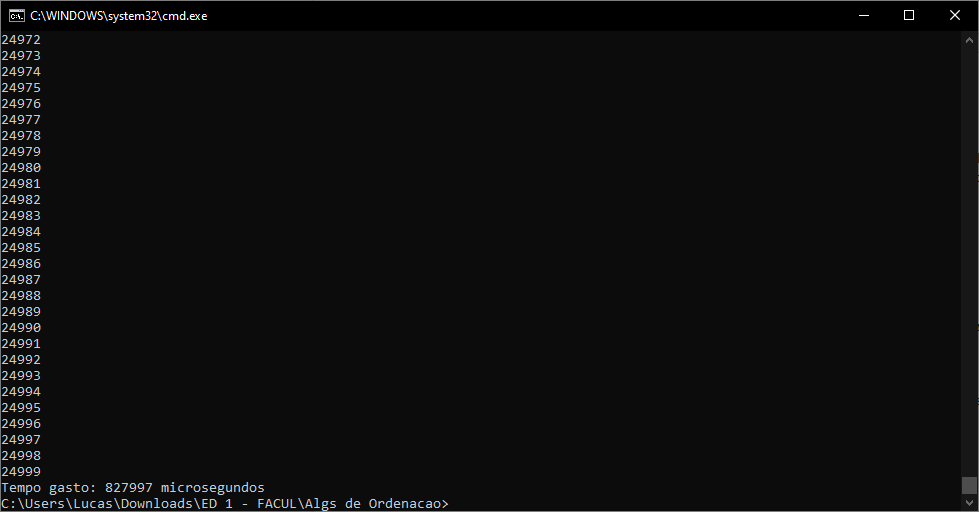
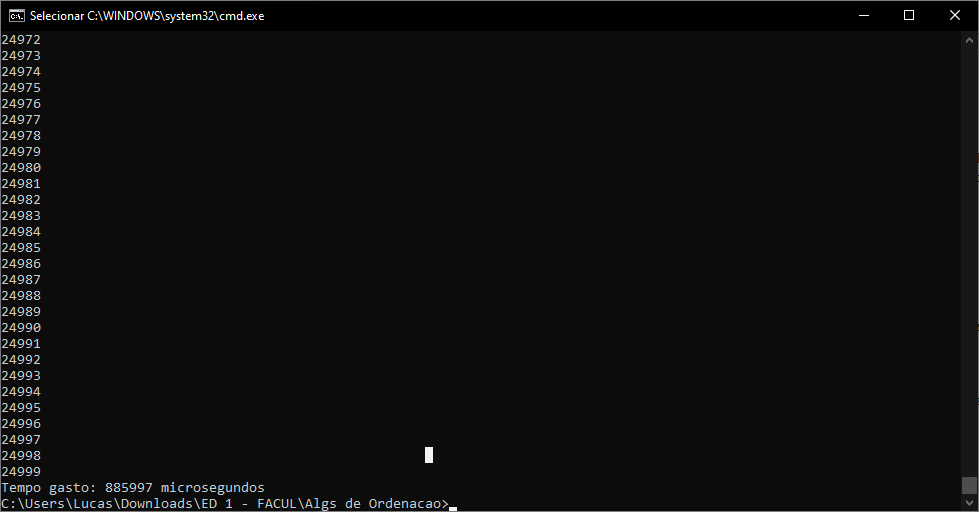
#### **desordenado125k**

1. 
2. 
3. 
4. 
5. 

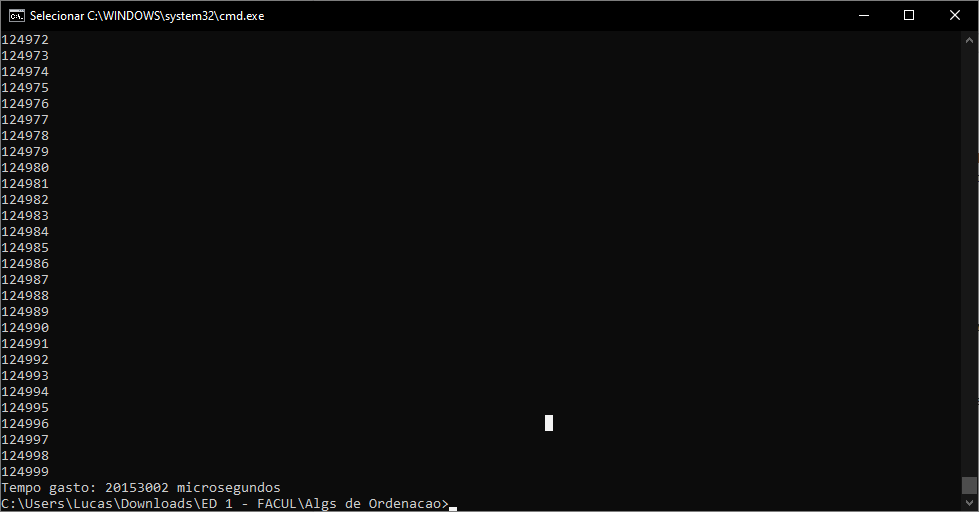
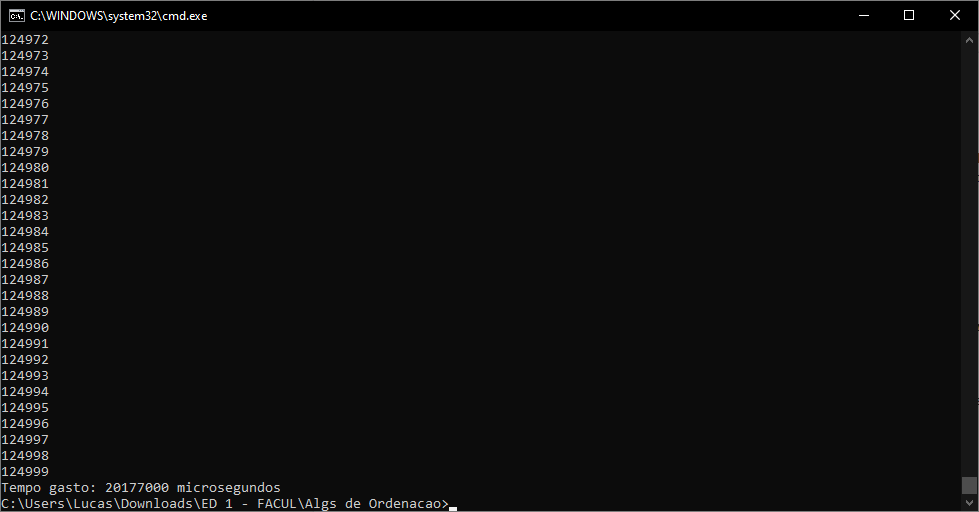
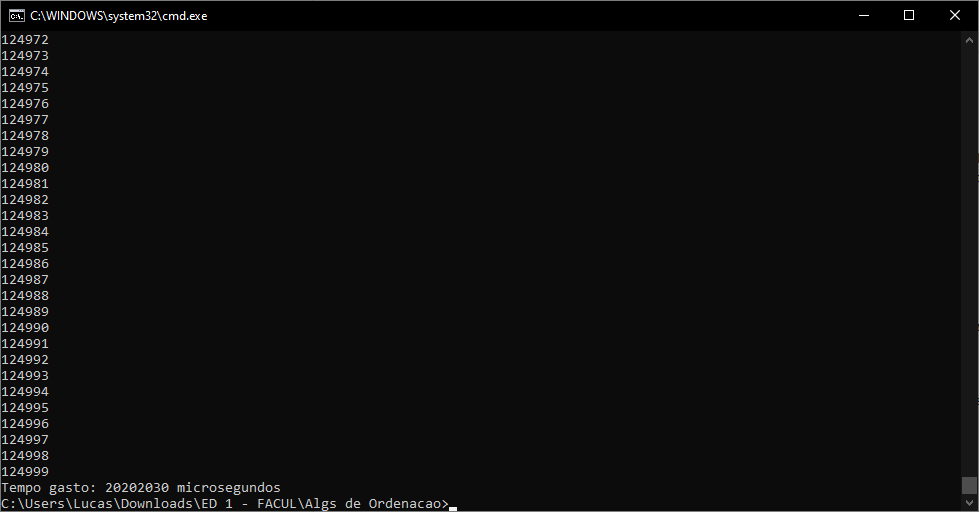
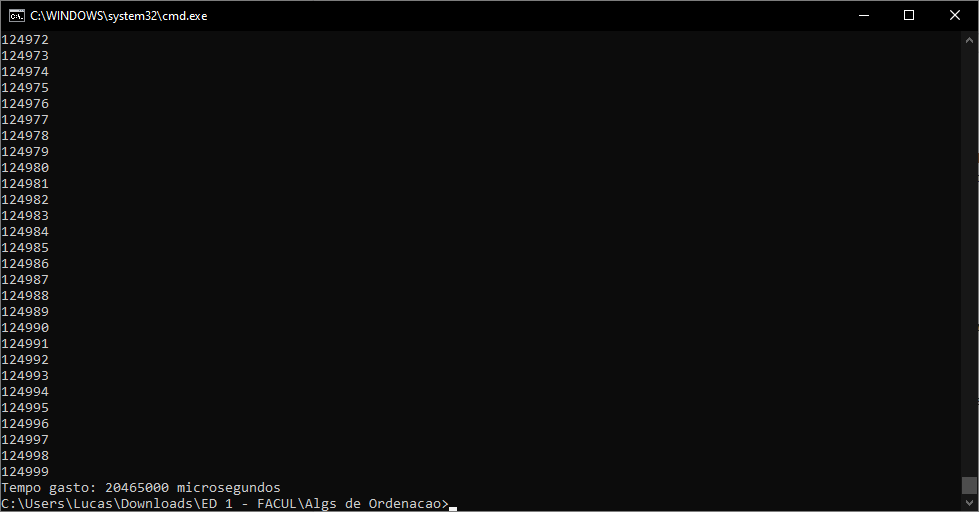
#### **ordenado5k**

1. 
2. 
3. 
4. 
5. 

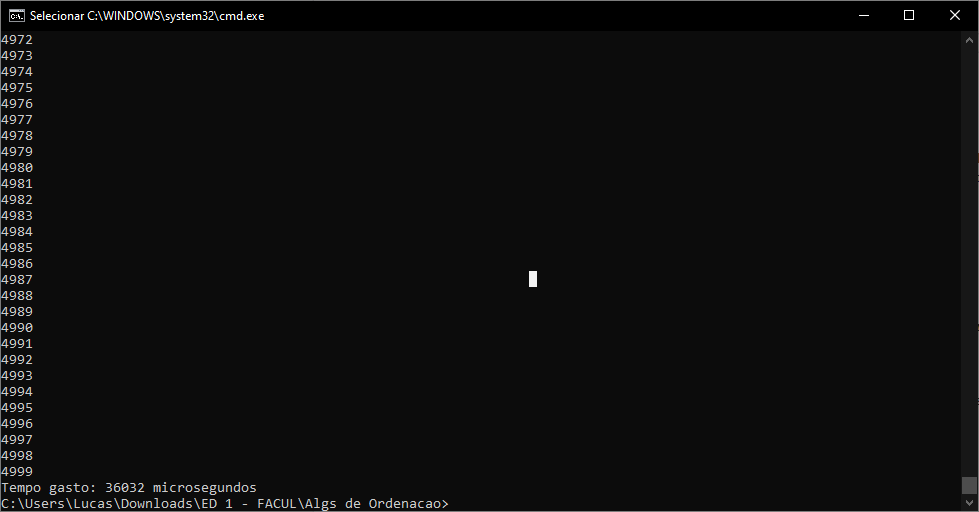
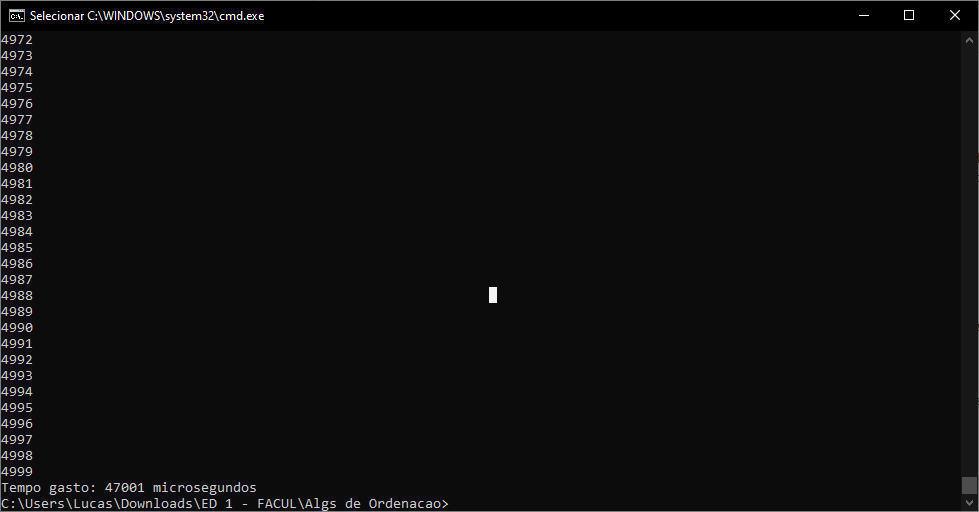
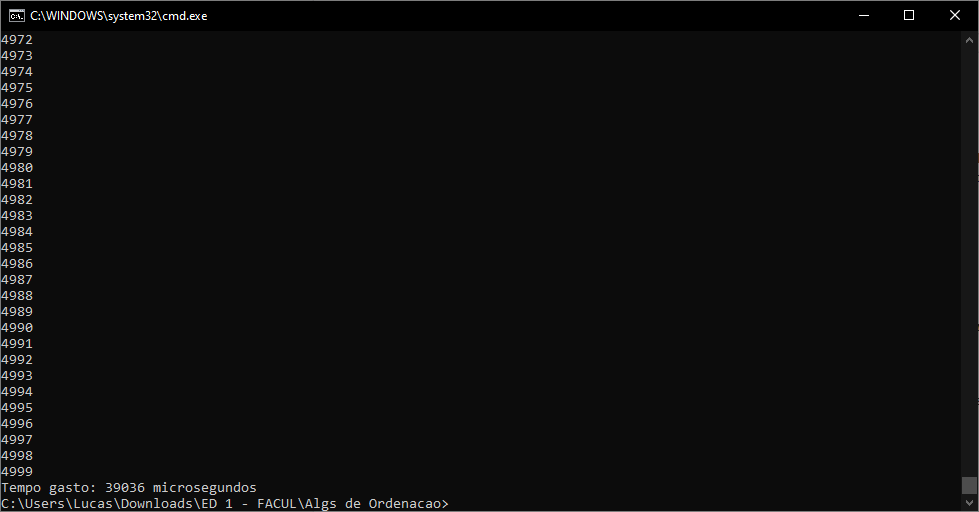
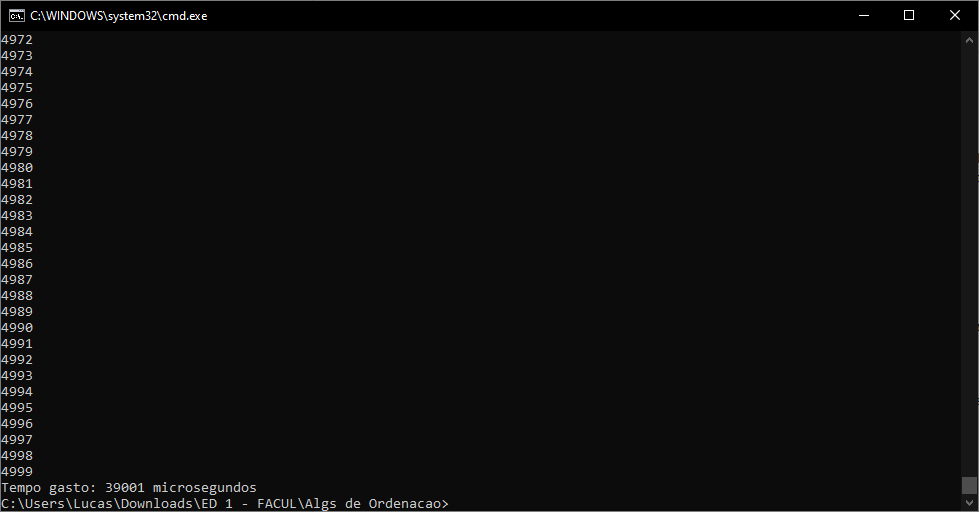
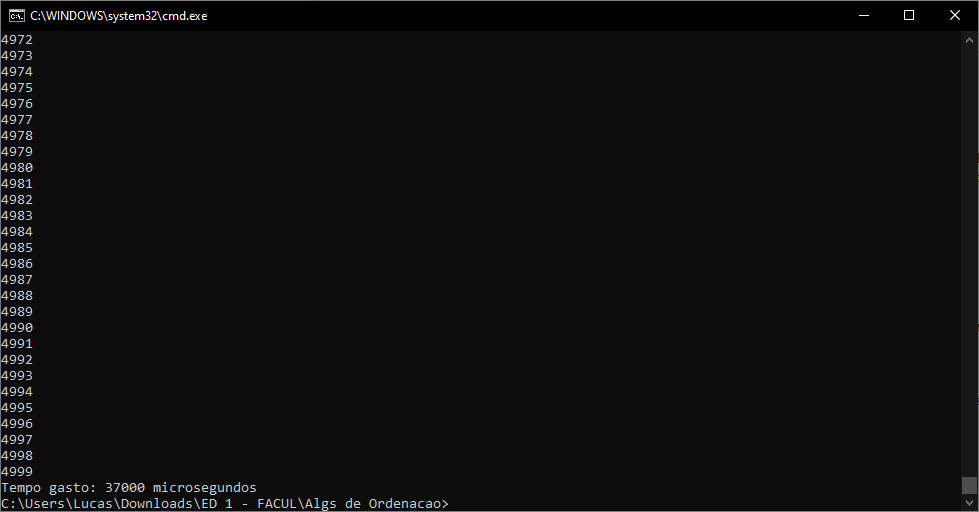
#### **ordenado25k**

1. 
2. 
3. 
4. 
5. 

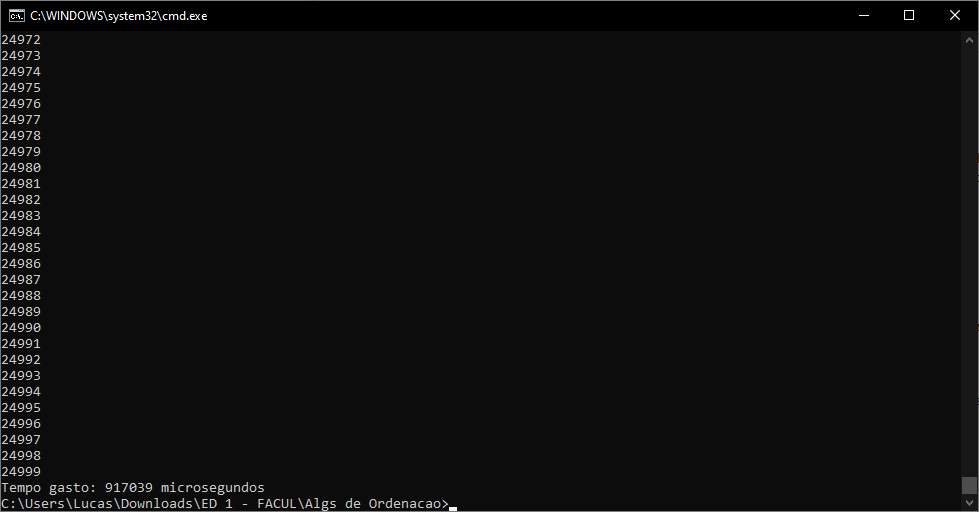
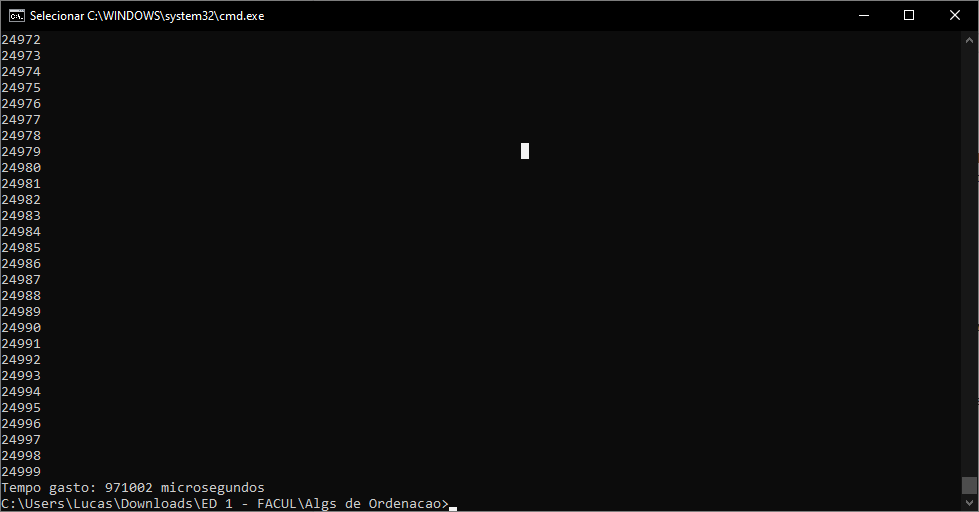
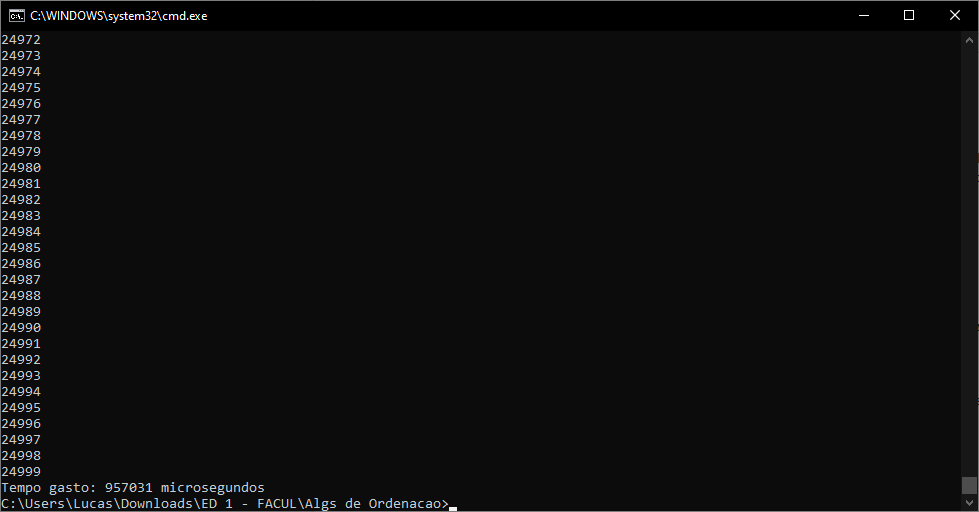
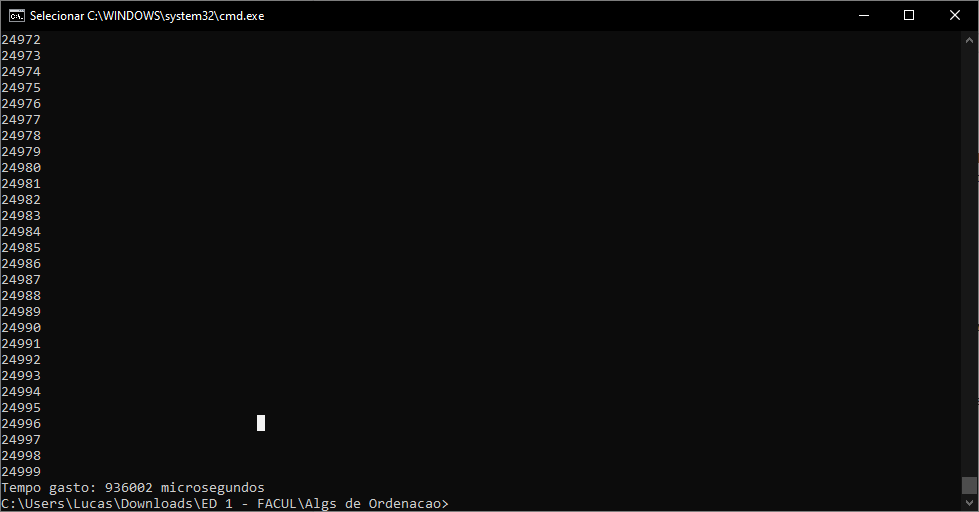
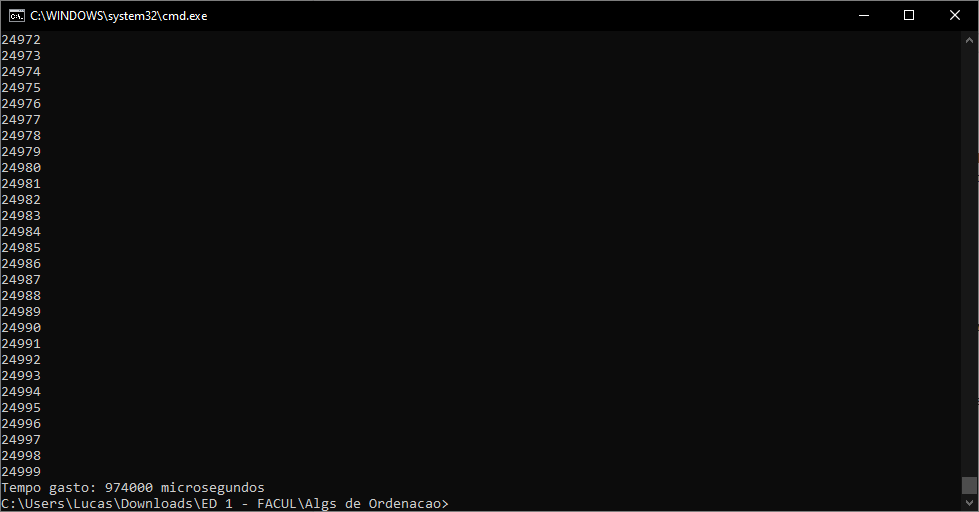
#### **ordenado125k**

1. 
2. 
3. 
4. 
5. 

#### **ordenadodesc5k**

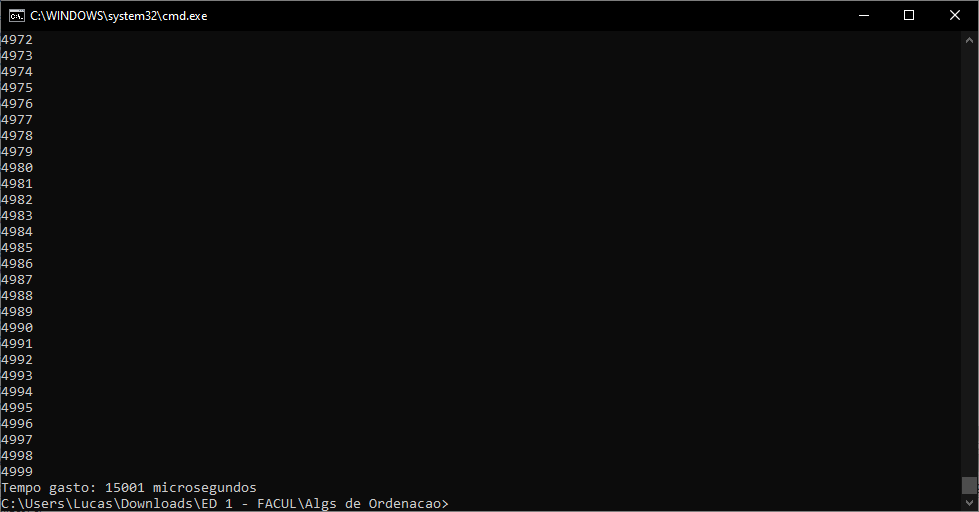
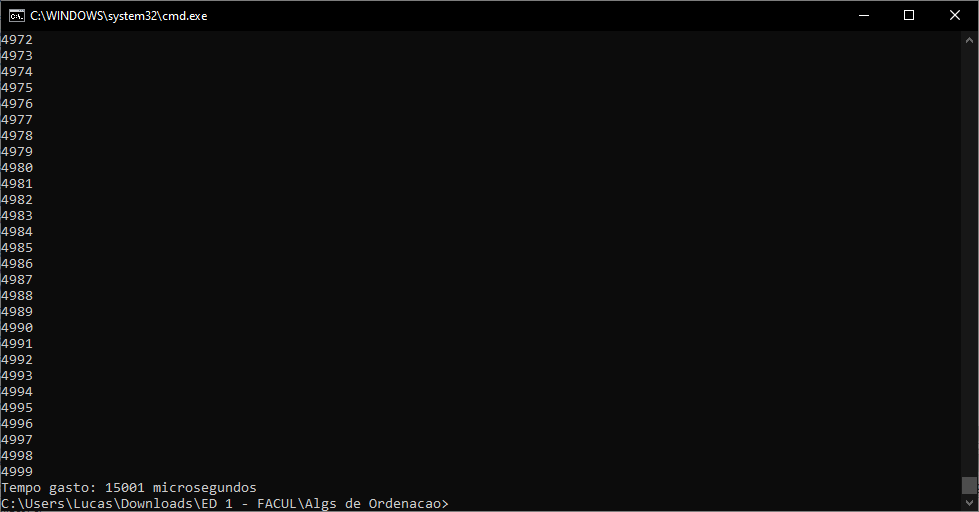
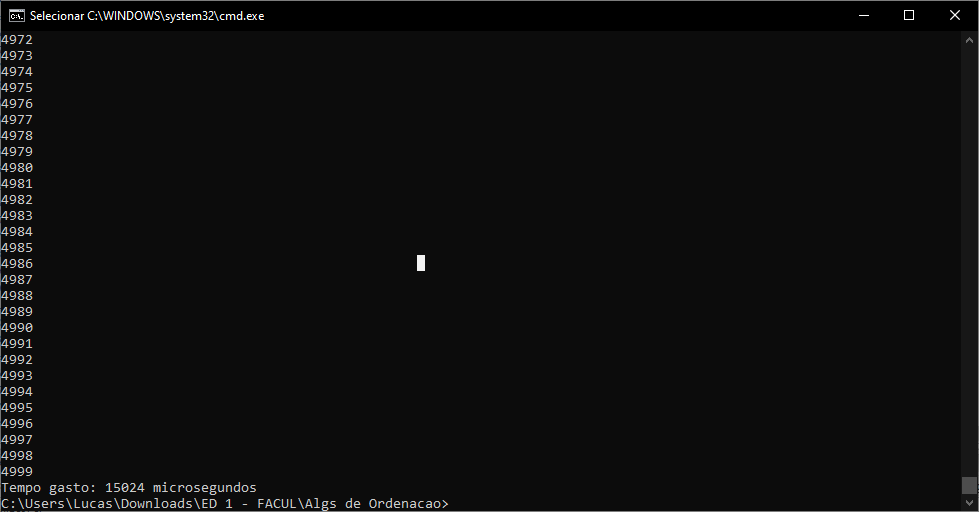
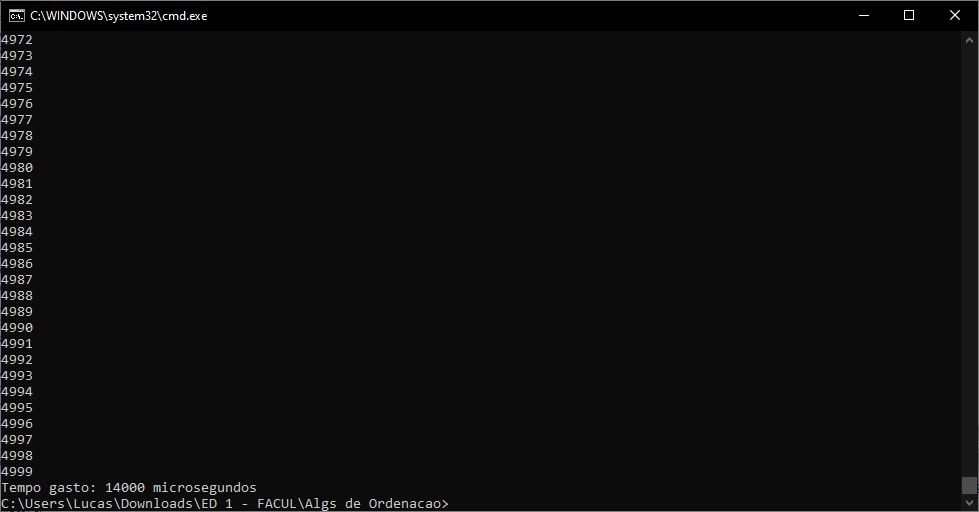
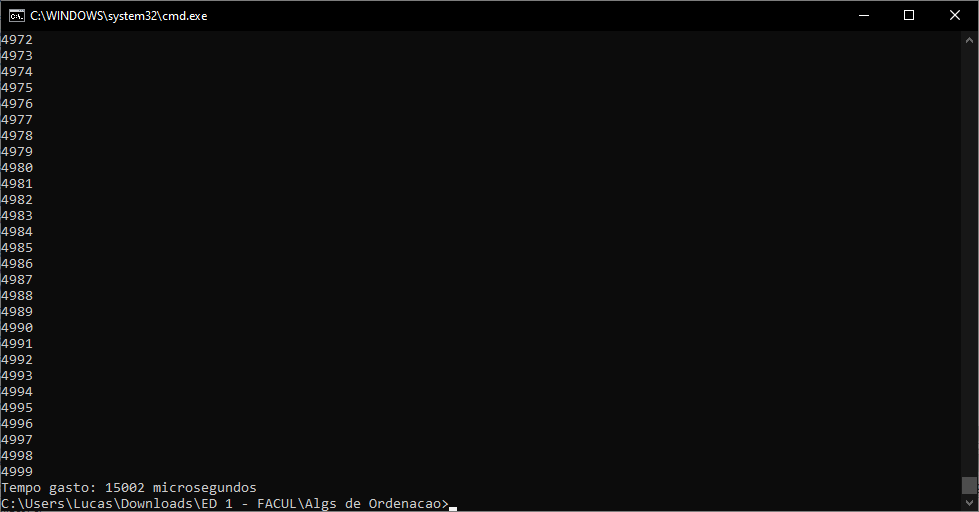
1. 
2. 
3. 
4. 
5. 

#### **ordenadodesc25k**

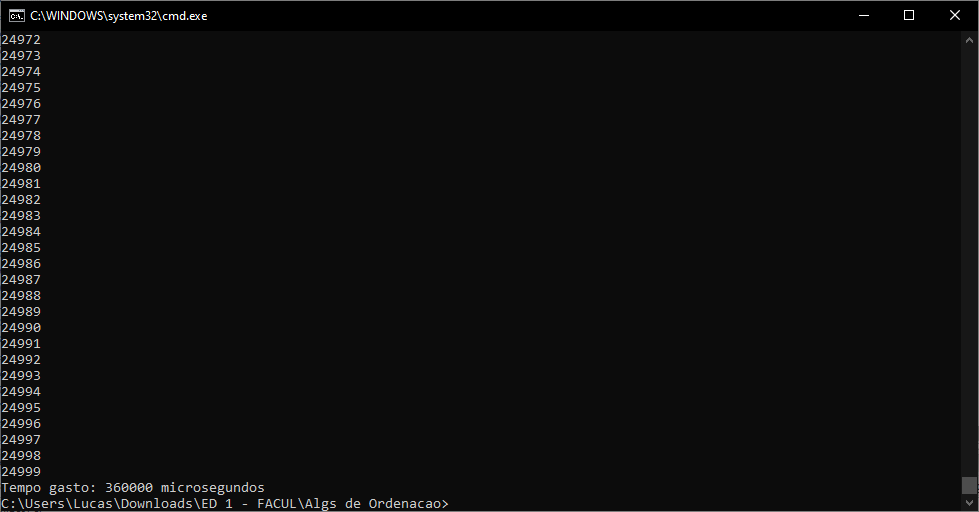
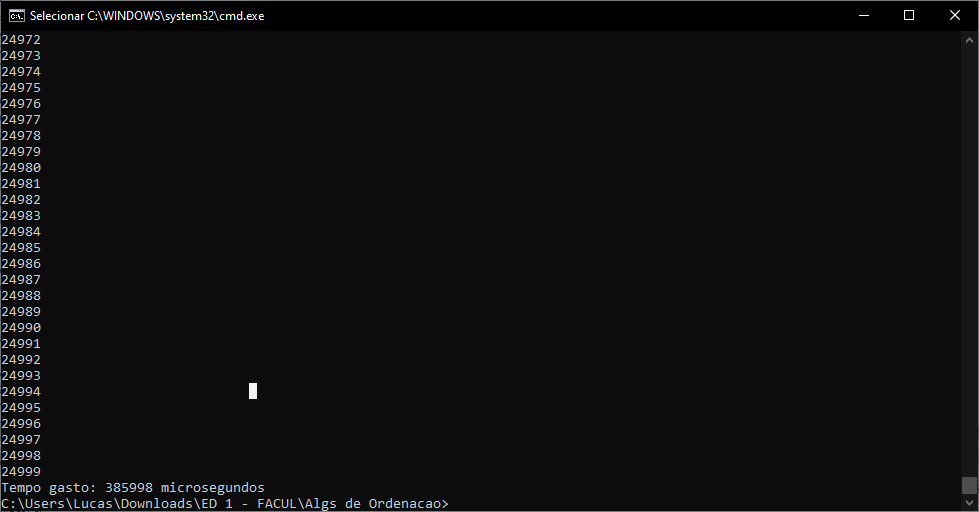
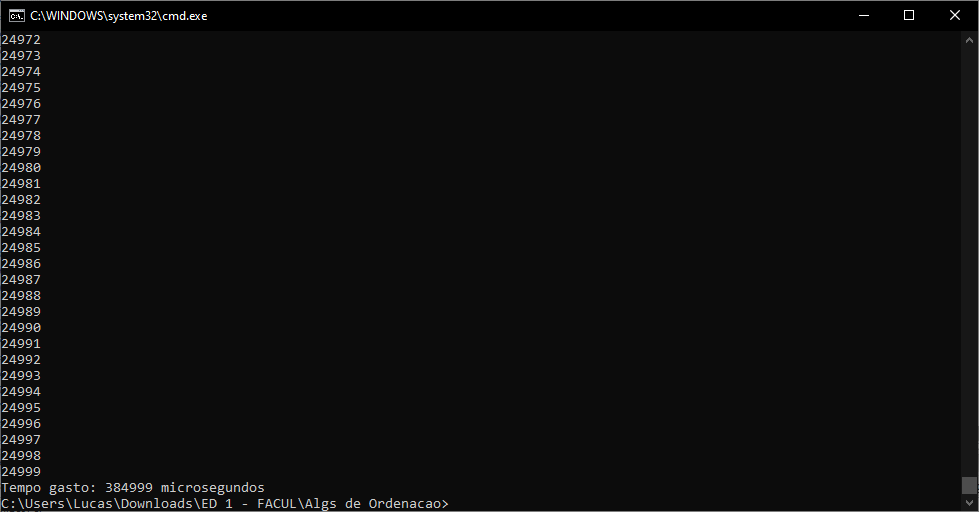
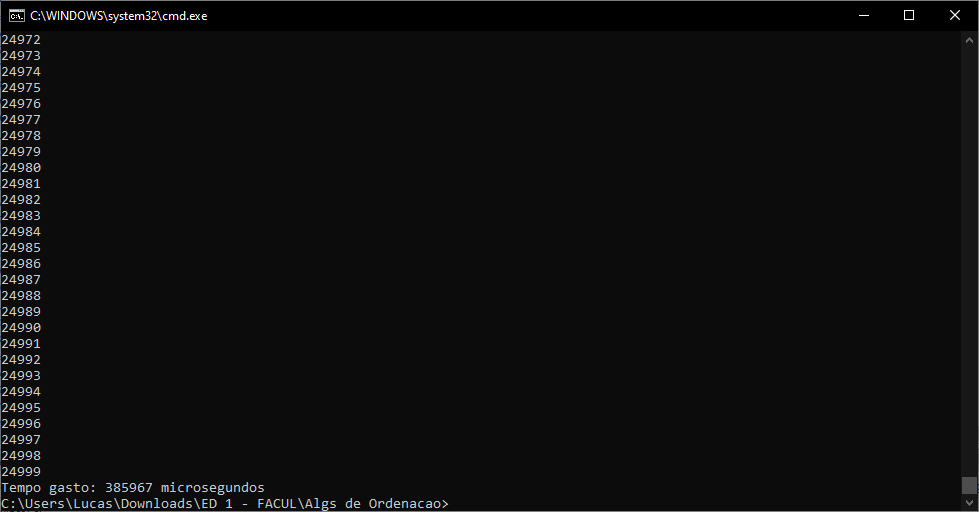
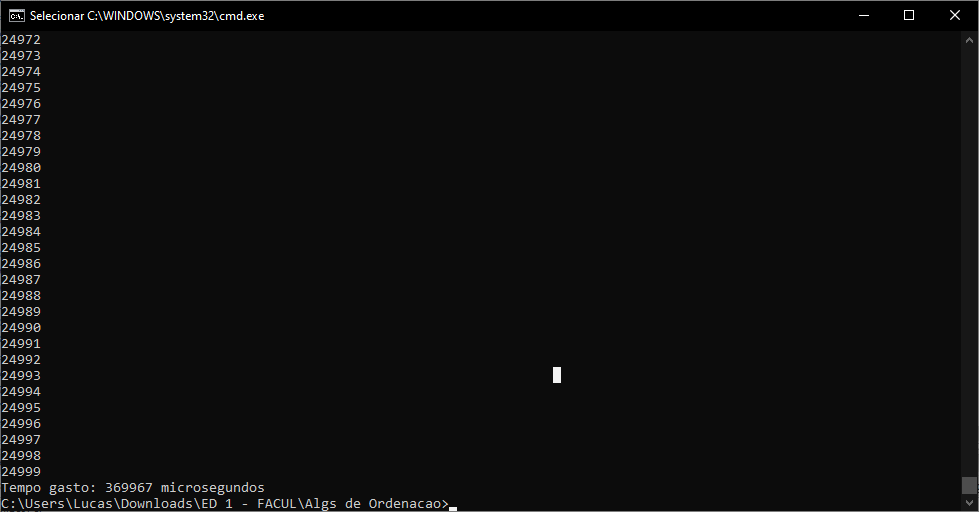
1. 
2. 
3. 
4. 
5. 

### **InsertionSort**

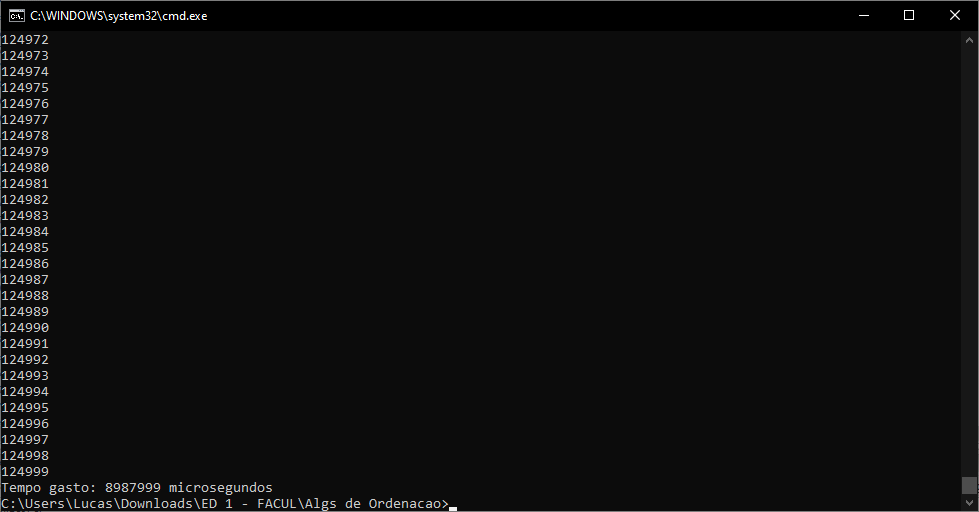
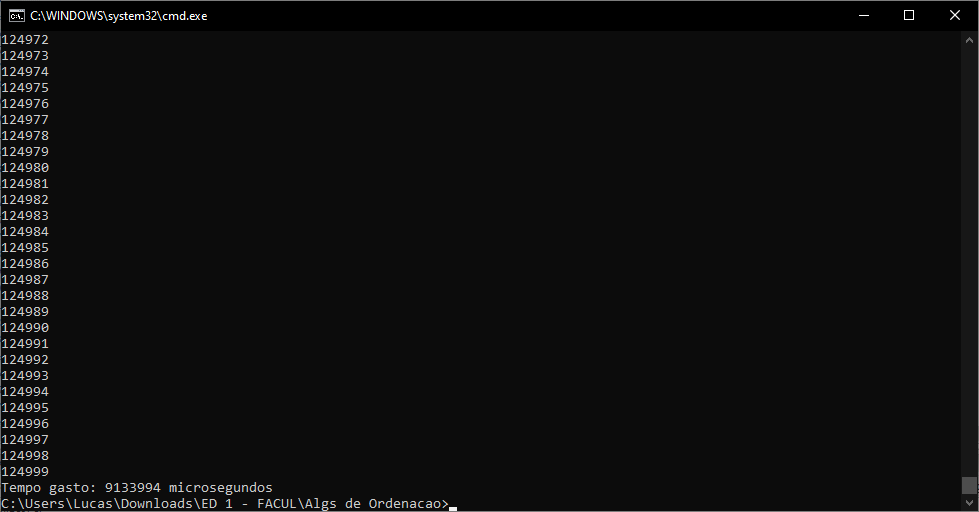
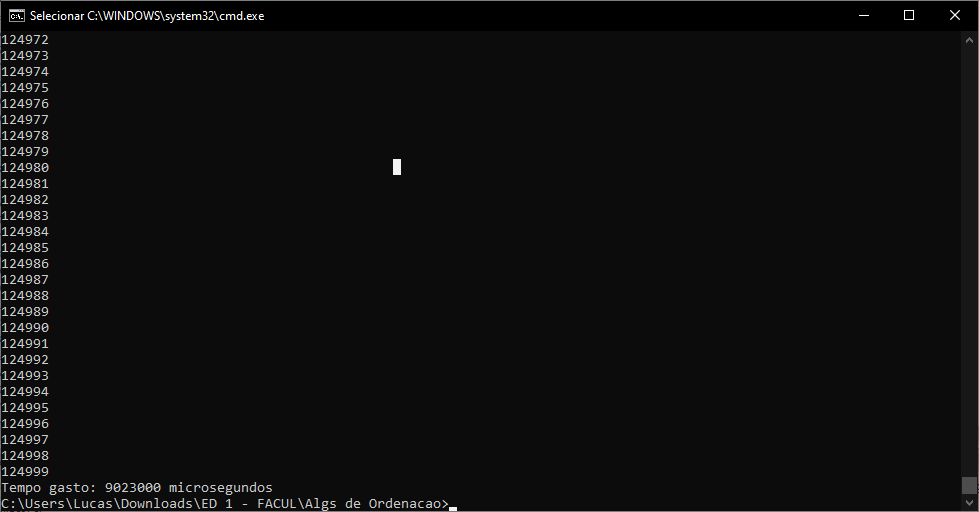
#### **desordenado5k**

1. 
2. 
3. 
4. 
5. 

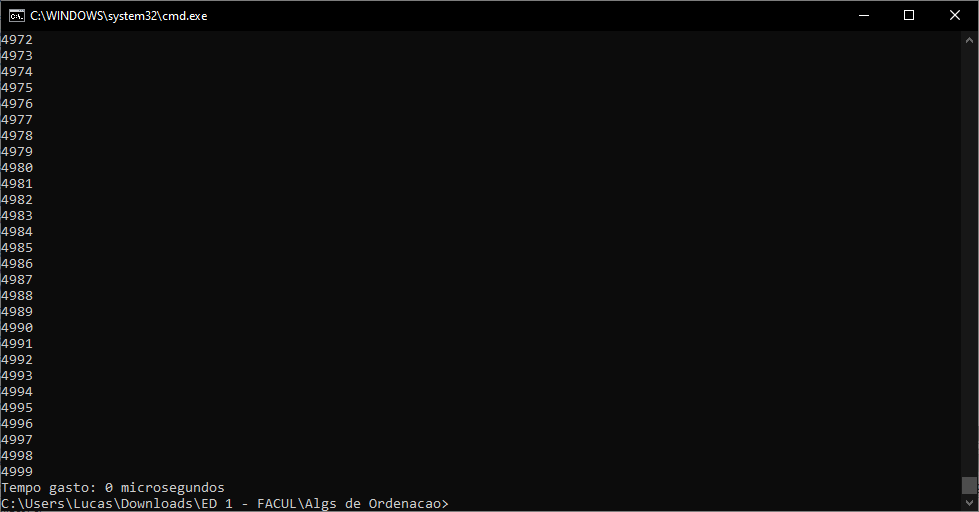
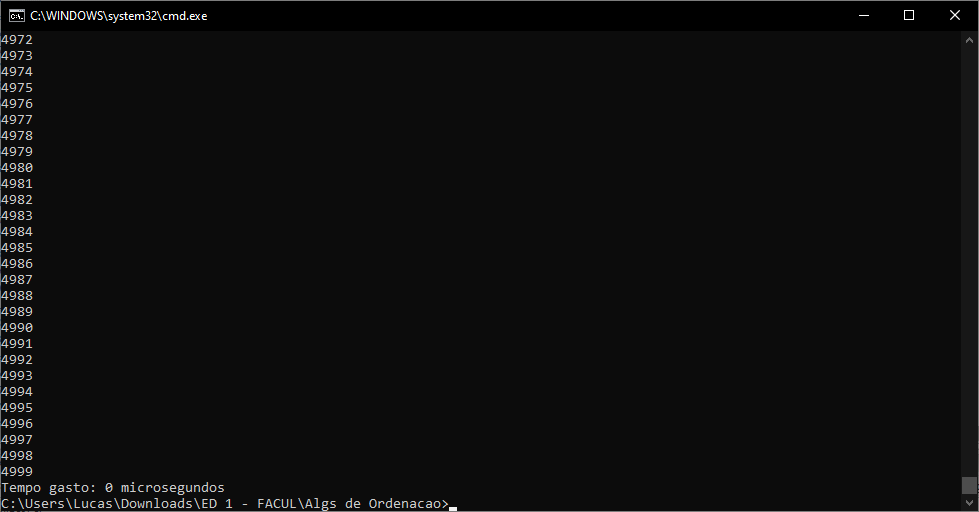
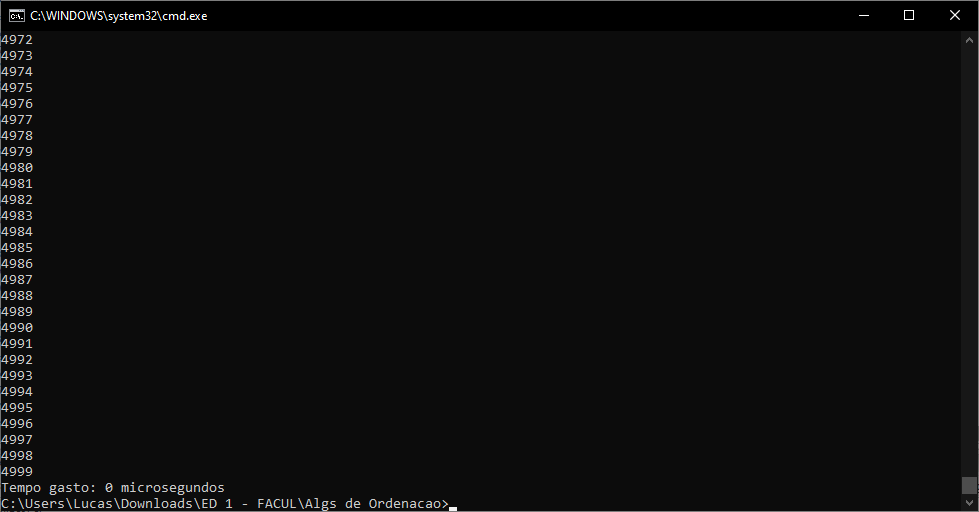
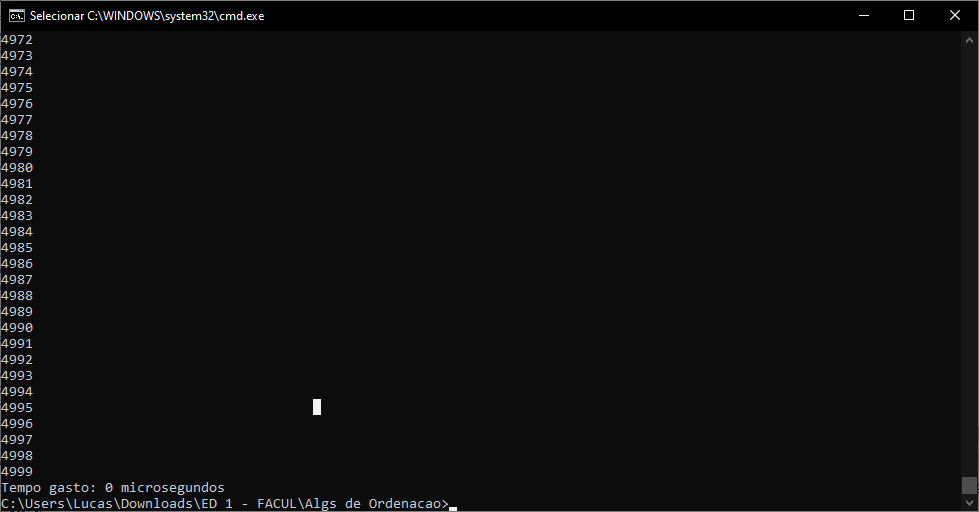
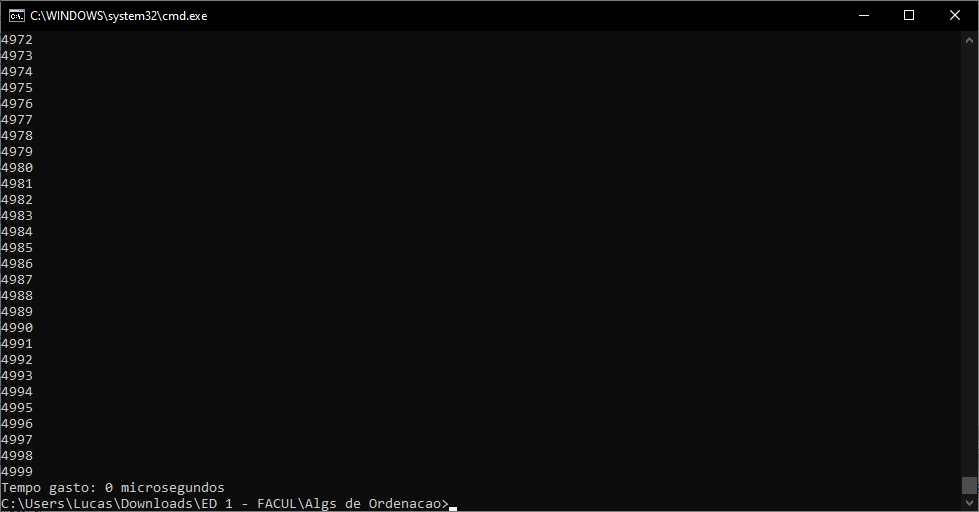
#### **desordenado25k**

1. 
2. 
3. 
4. 
5. 

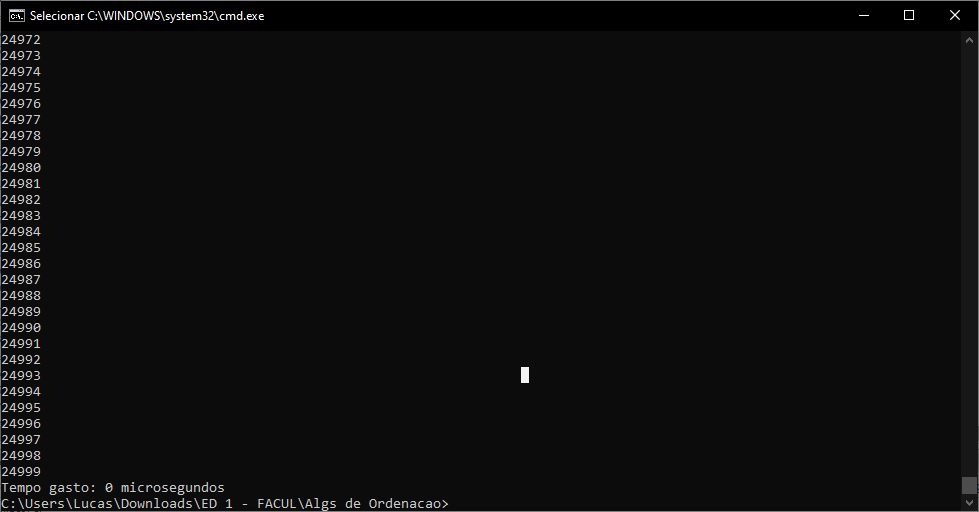
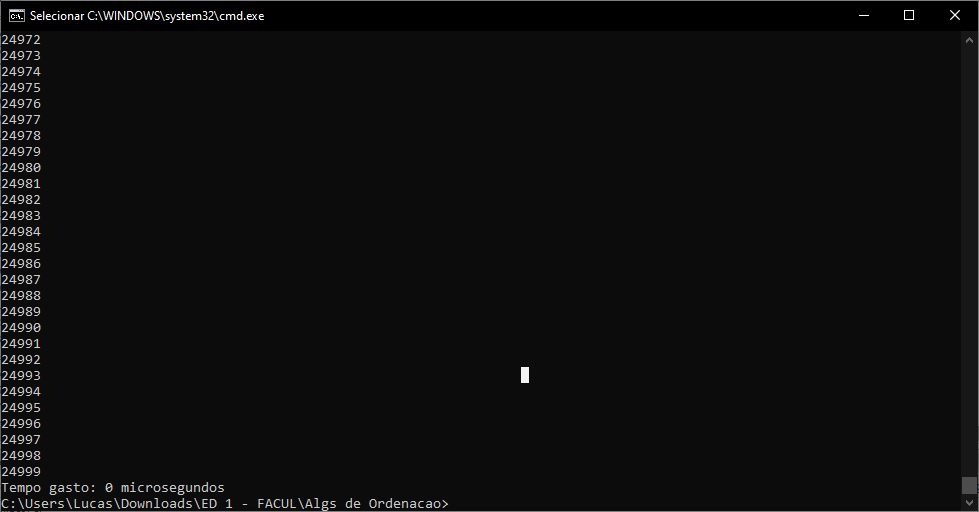
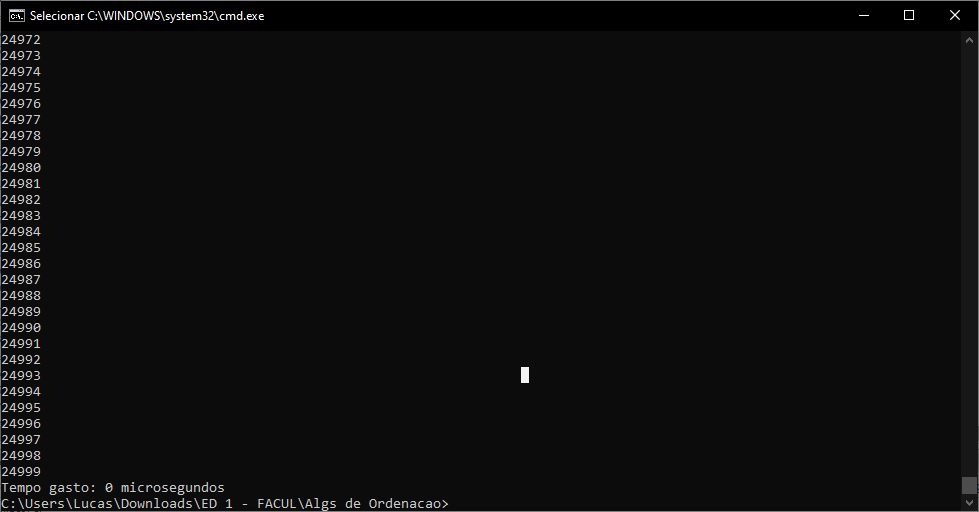
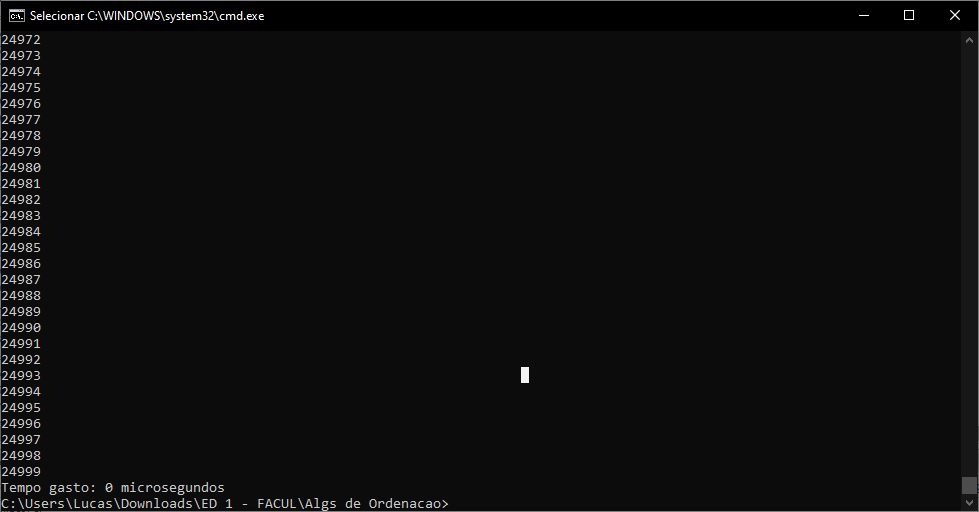
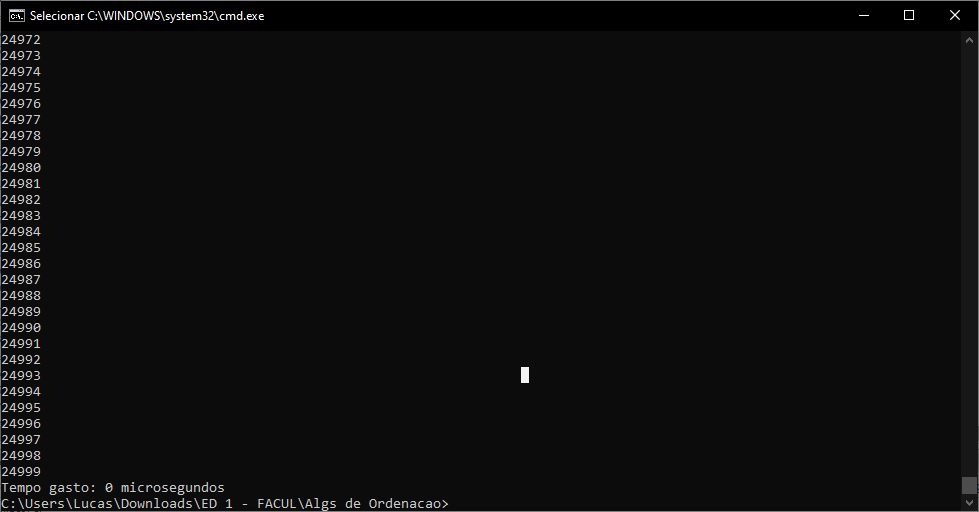
#### **desordenado125k**

1. 
2. 
3. 
4. 
5. 

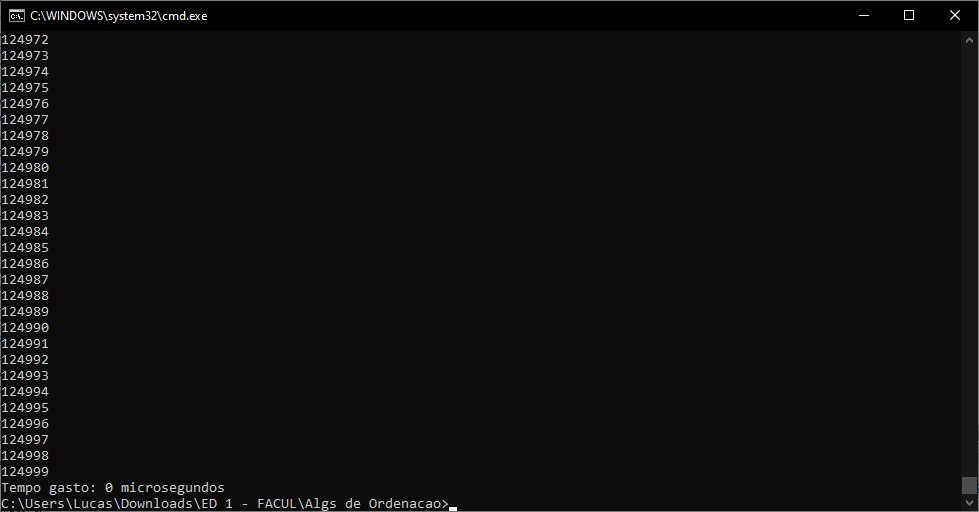
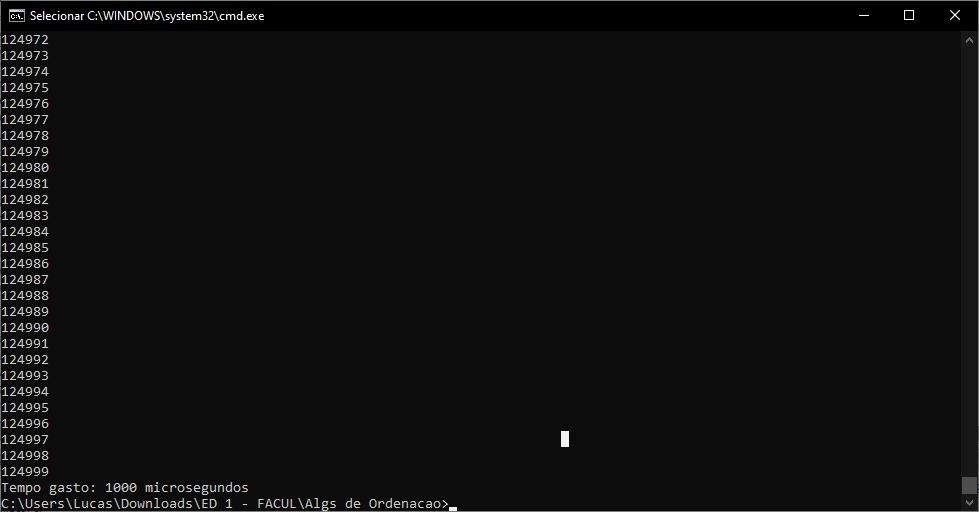
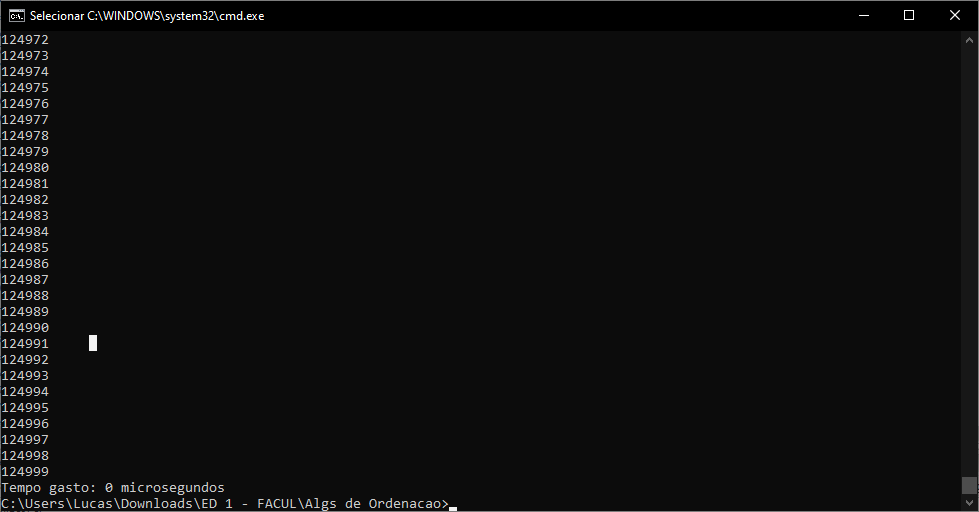
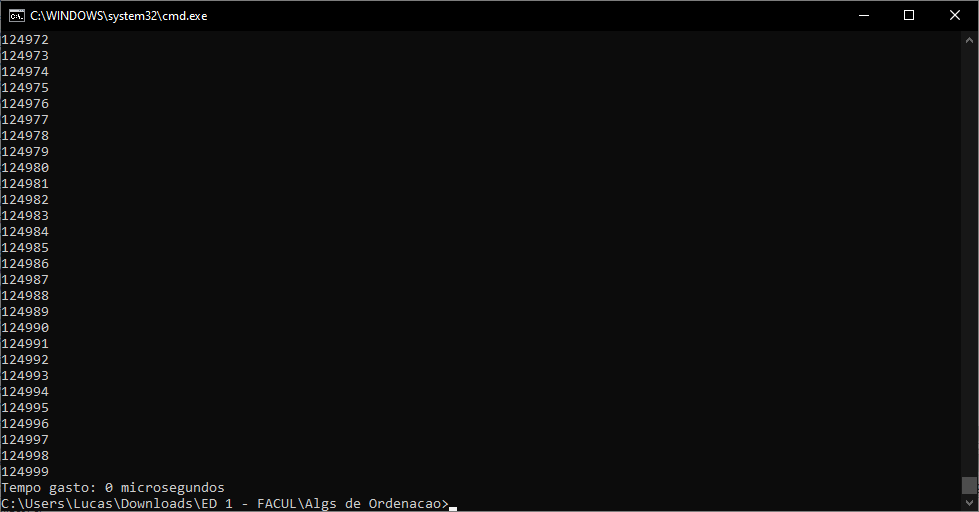
#### **ordenado5k**

1. 
2. 
3. 
4. 
5. 

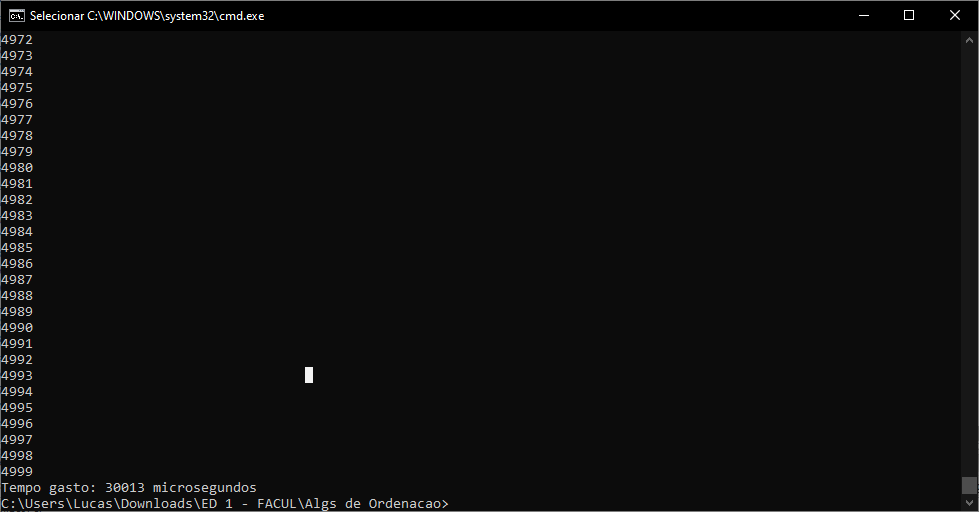
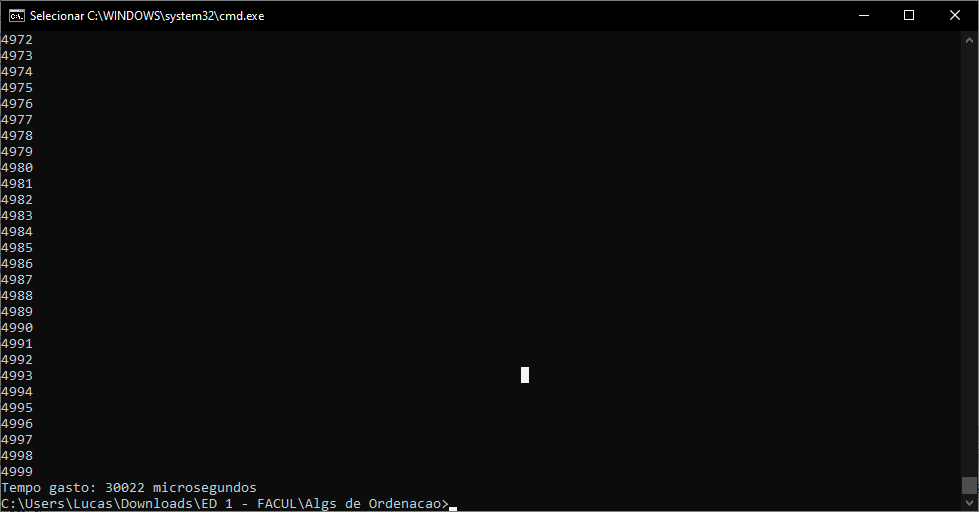
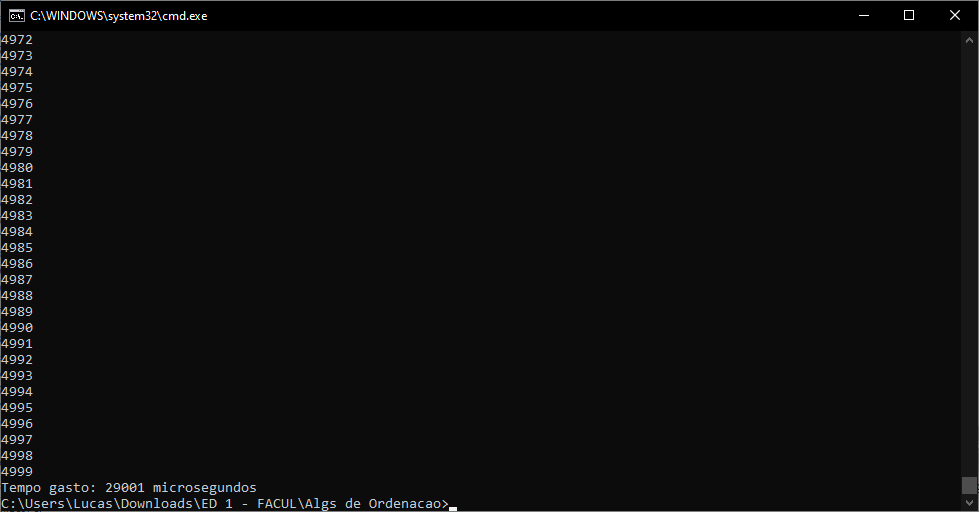
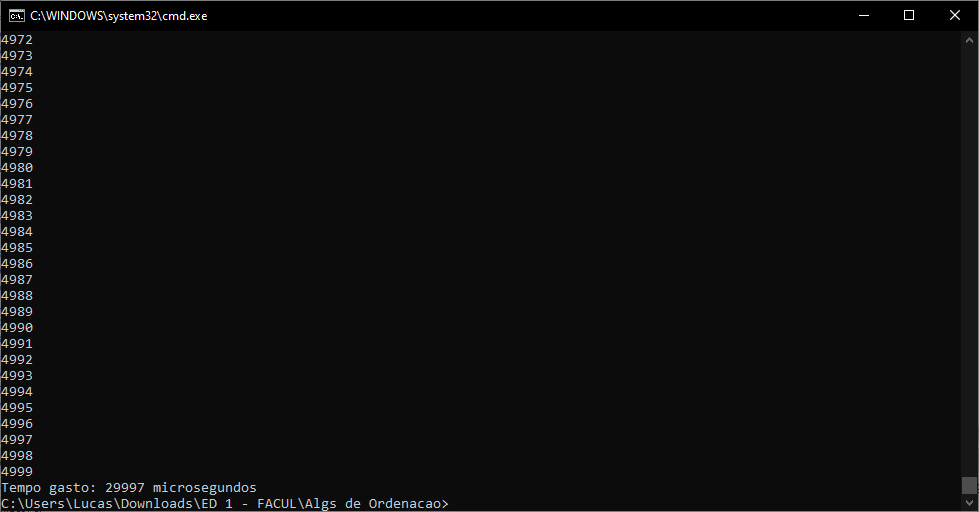
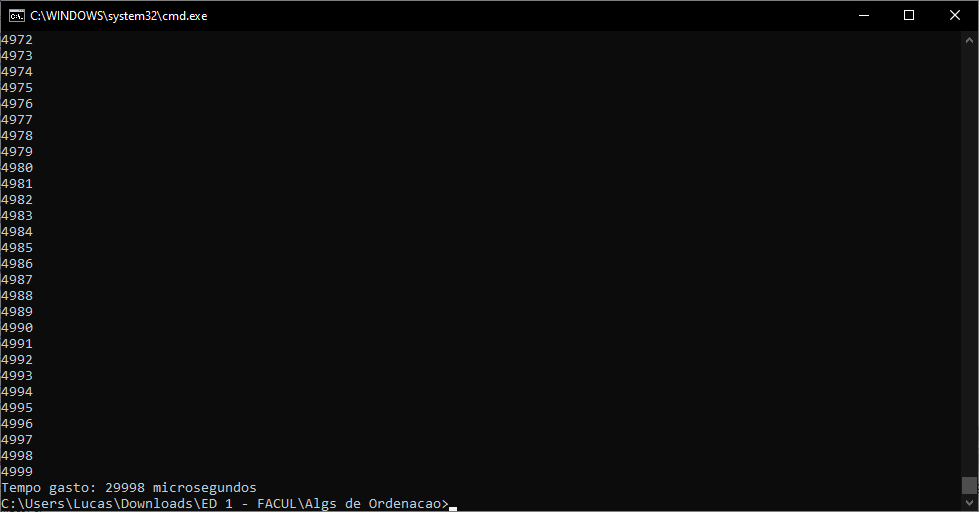
#### **ordenado25k**

1. 
2. 
3. 
4. 
5. 

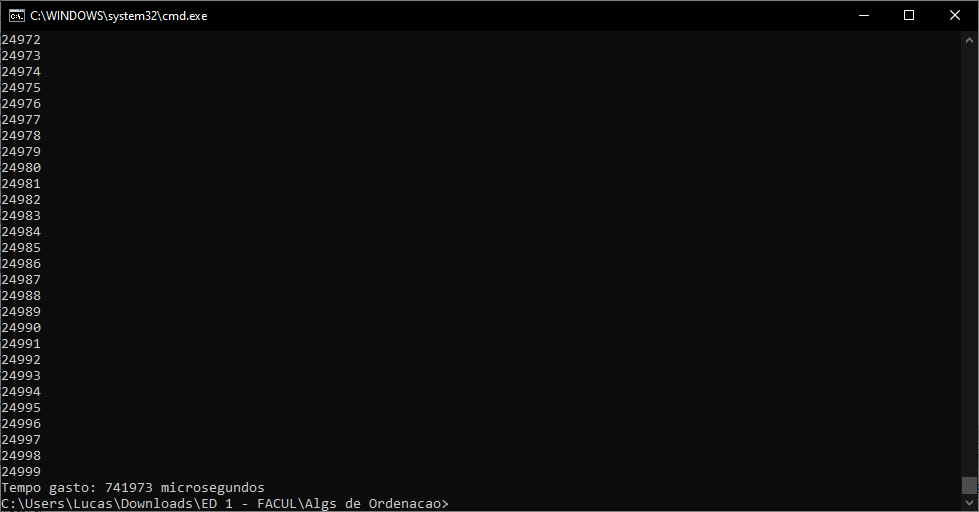
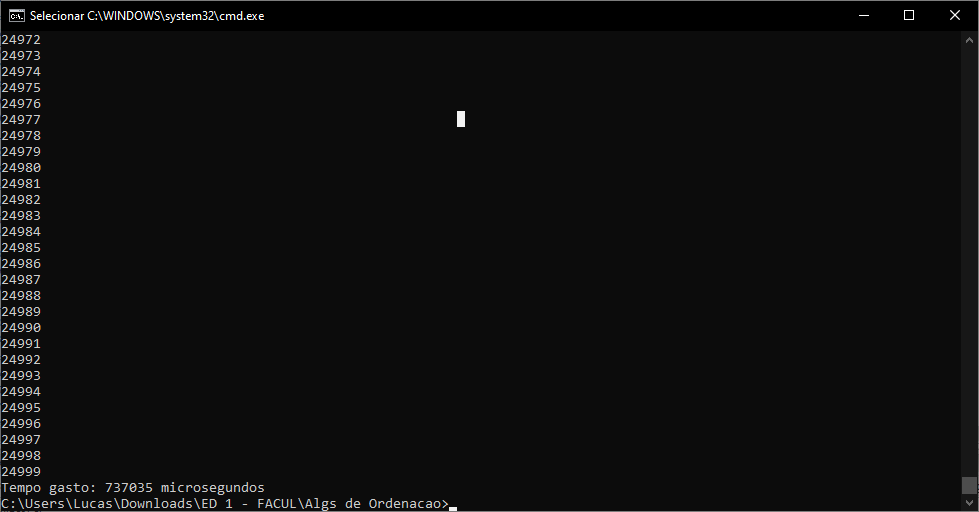
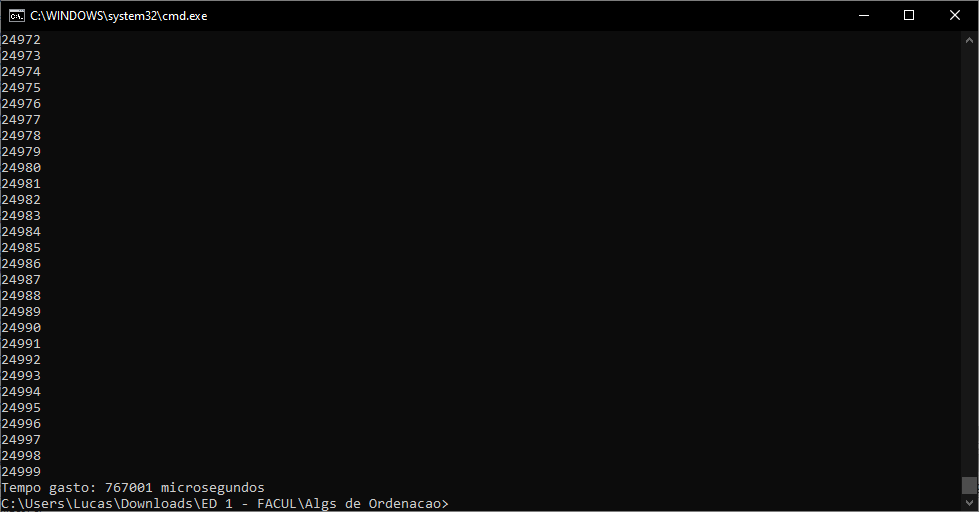
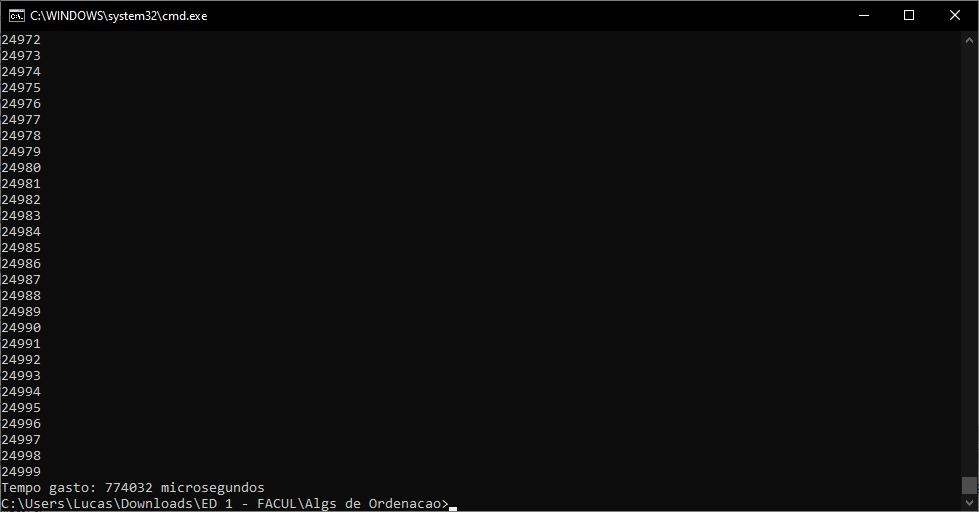
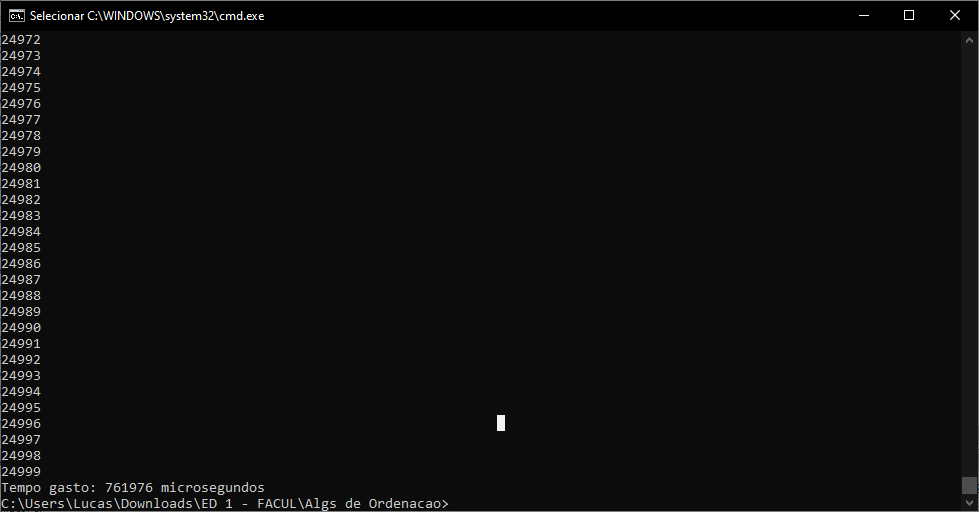
#### **ordenado125k**

1. 
2. 
3. 
4. 
5. 

#### **ordenadodesc5k**

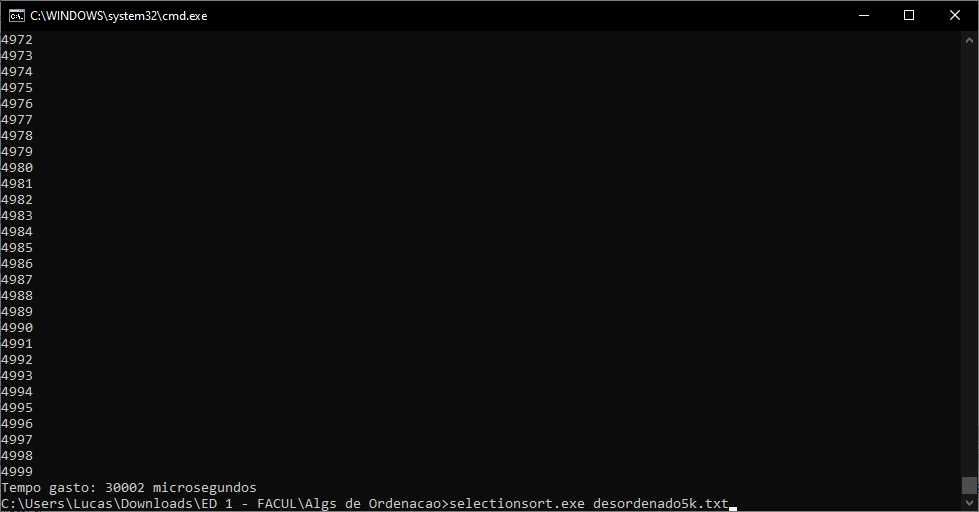
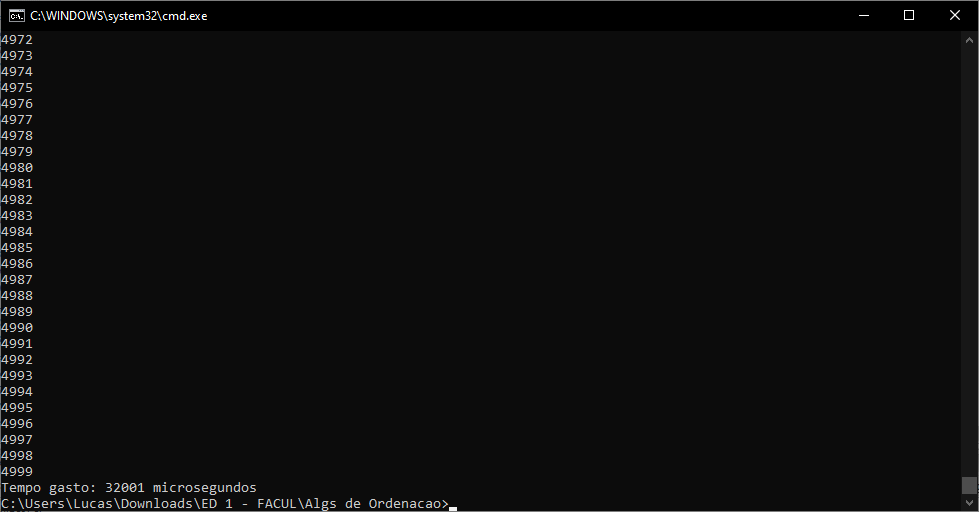
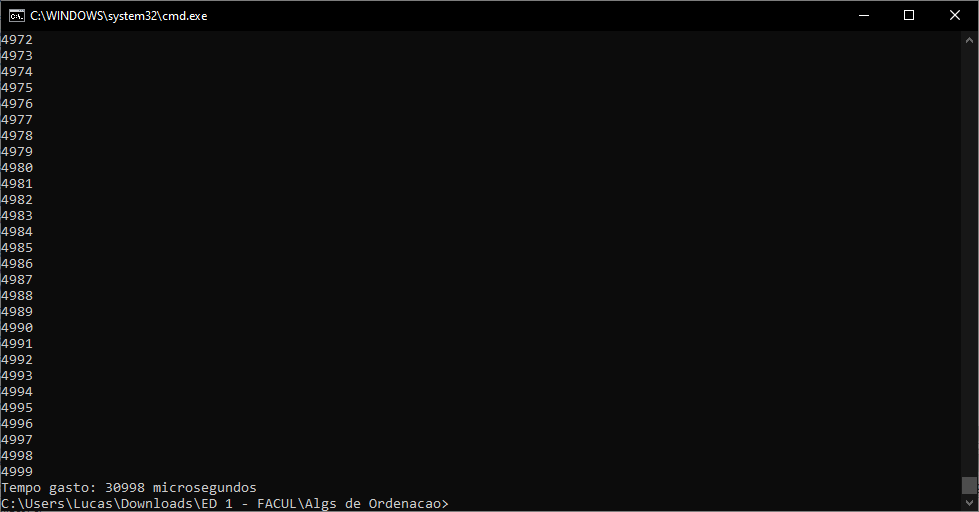
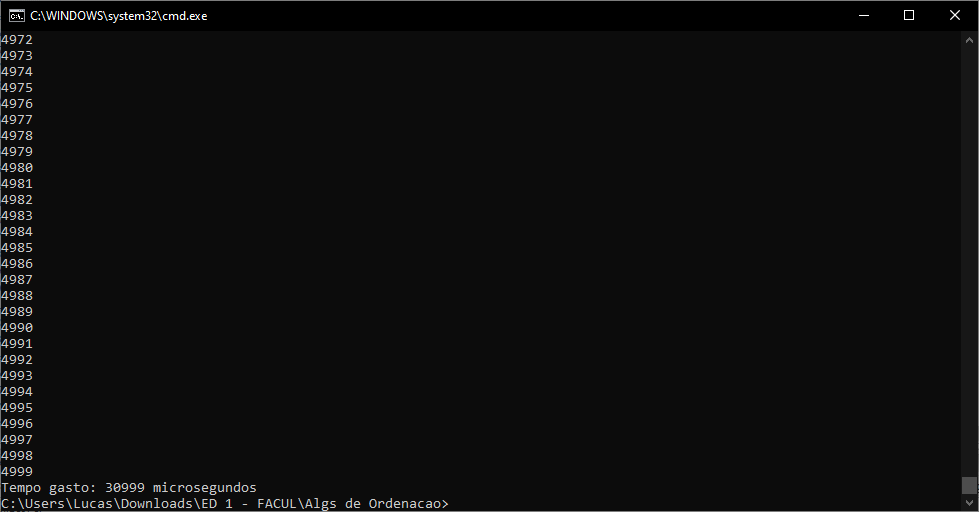
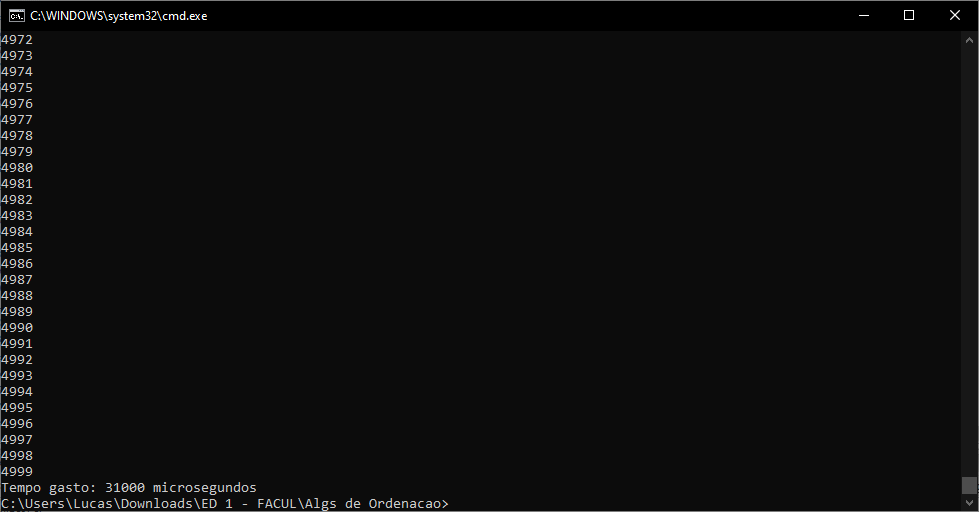
1. 
2. 
3. 
4. 
5. 

#### **ordenadodesc25k**

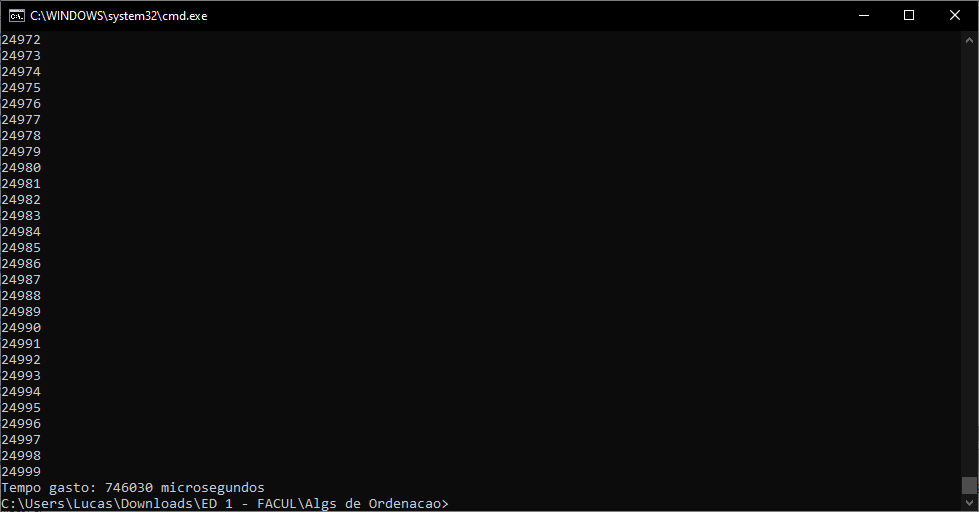
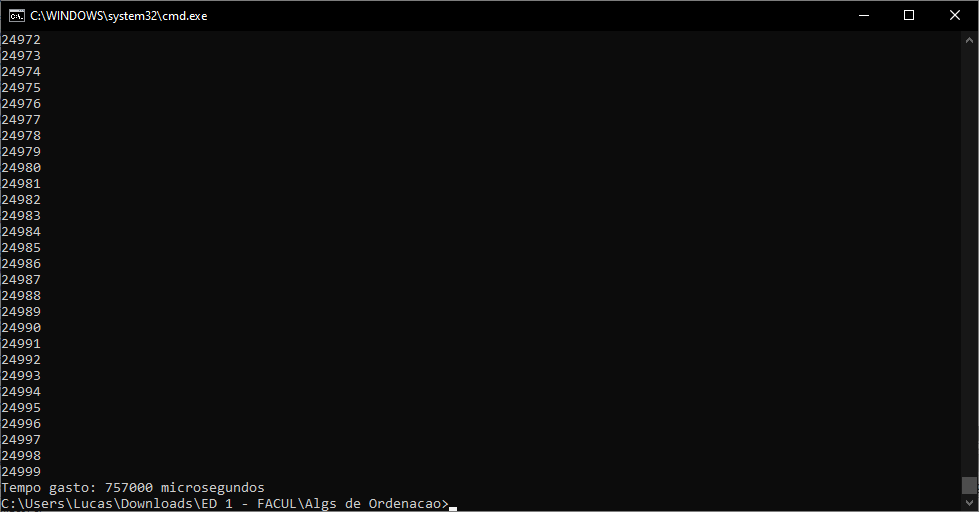
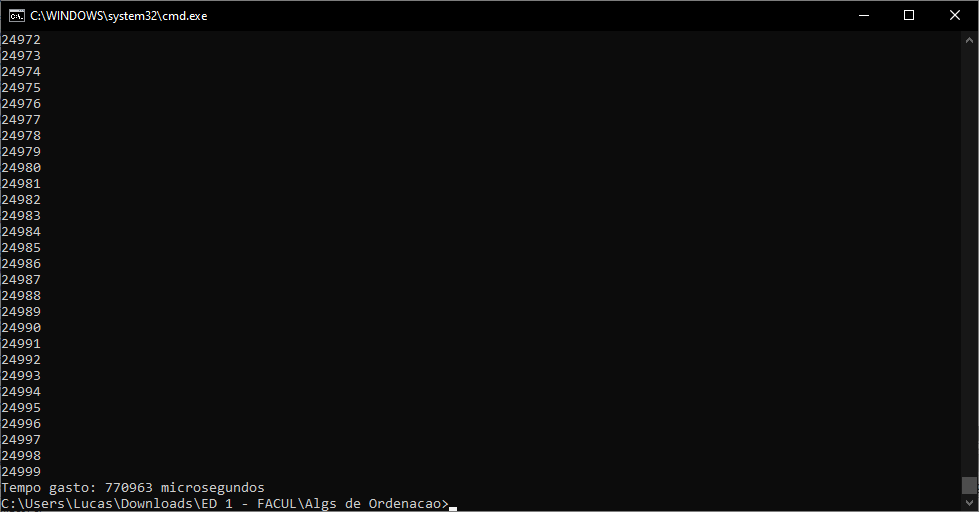
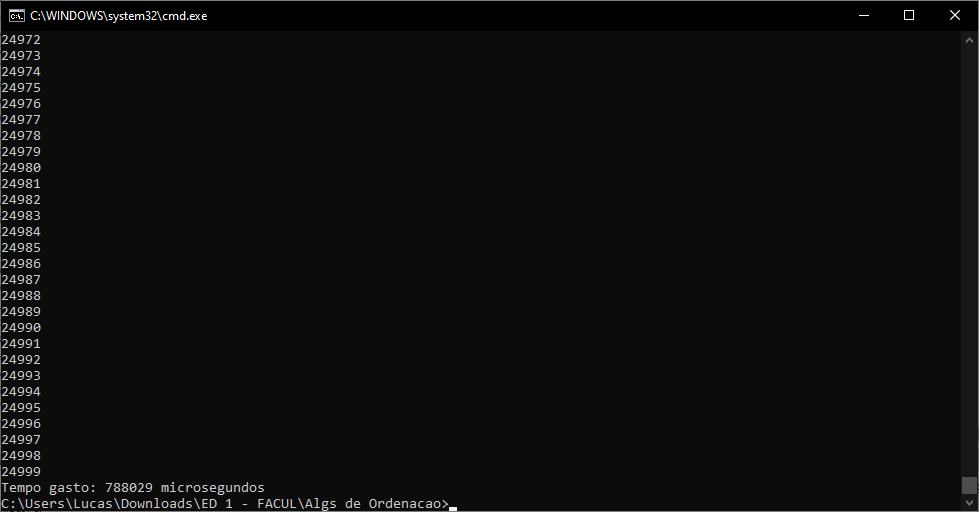
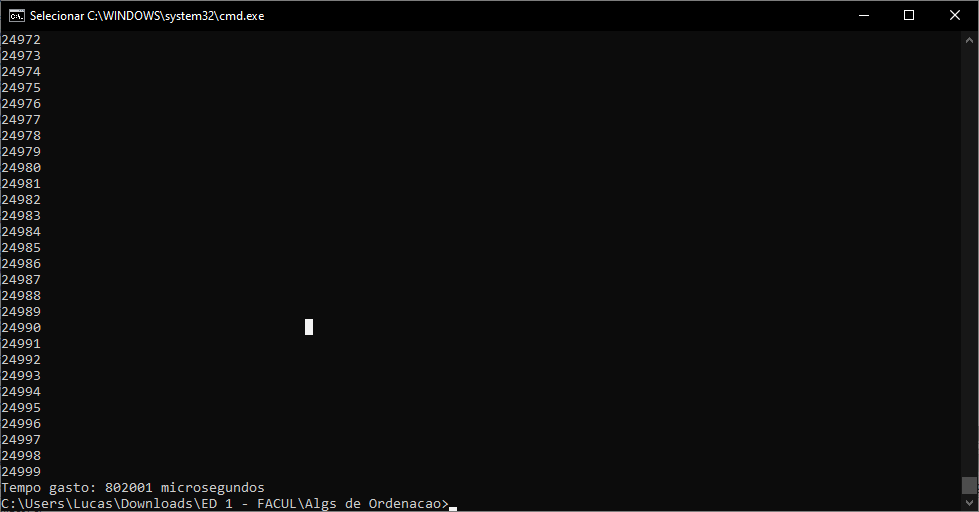
1. 
2. 
3. 
4. 
5. 

### **SelectionSort**

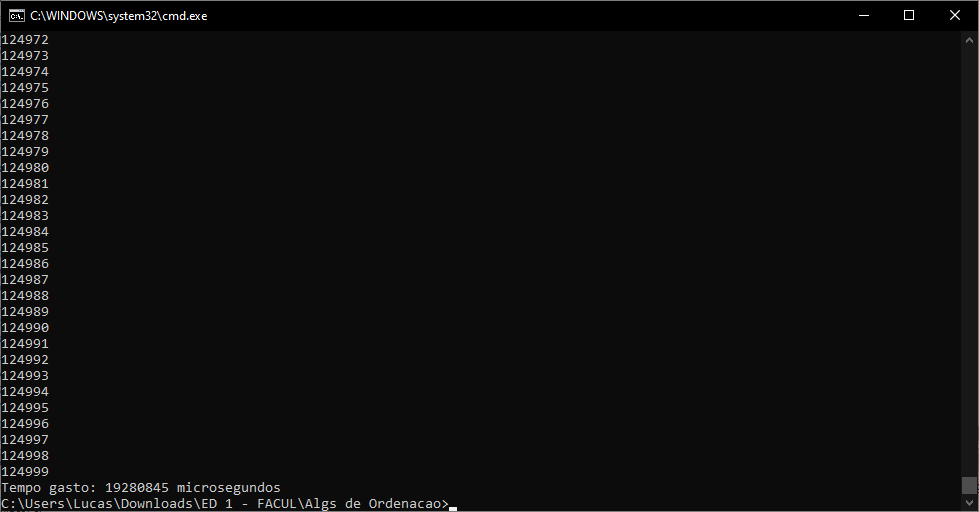
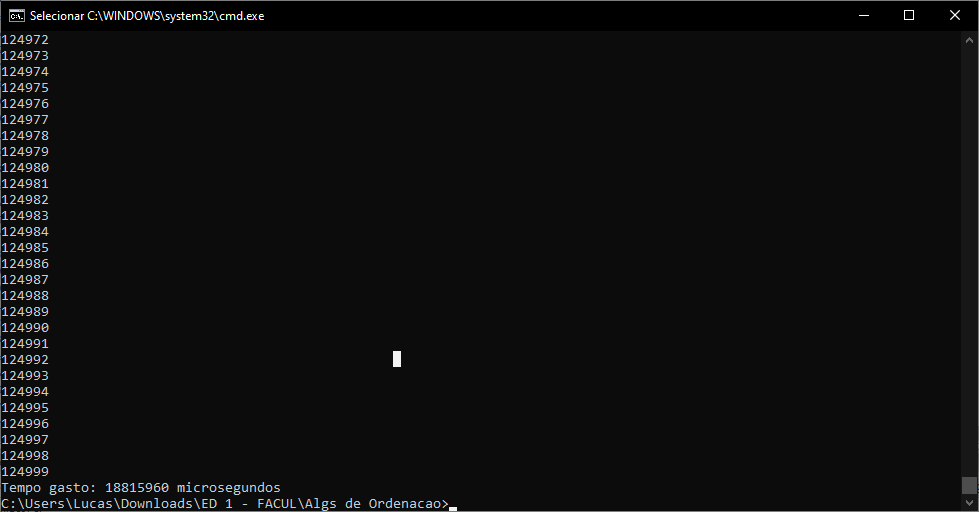
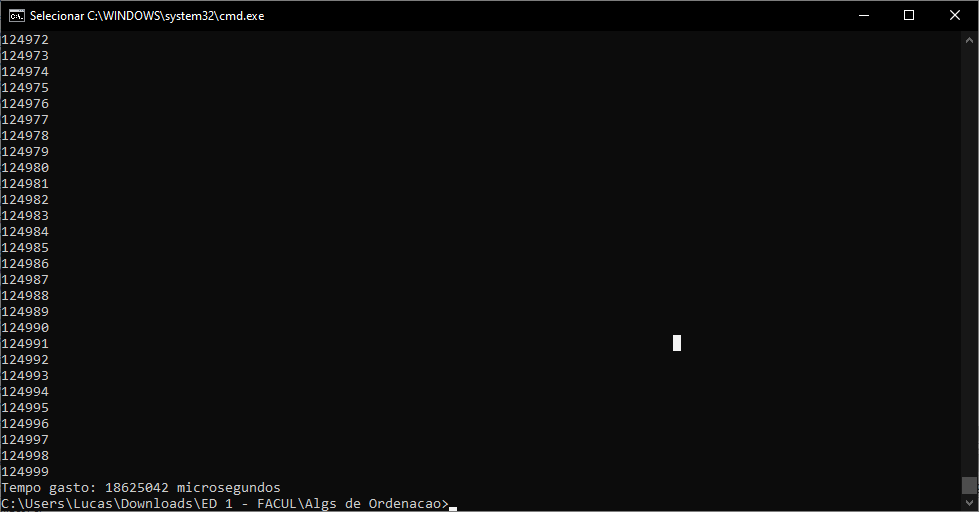
#### **desordenado5k**

1. 
2. 
3. 
4. 
5. 

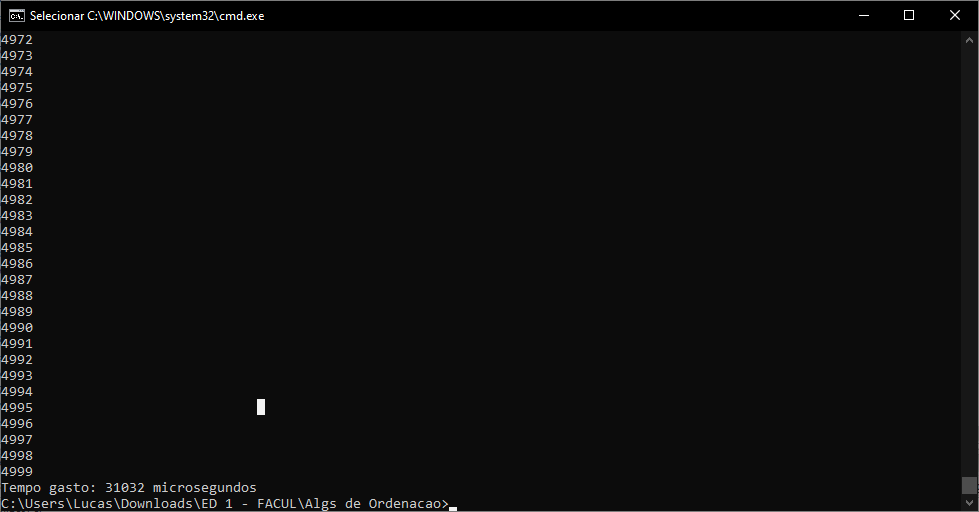
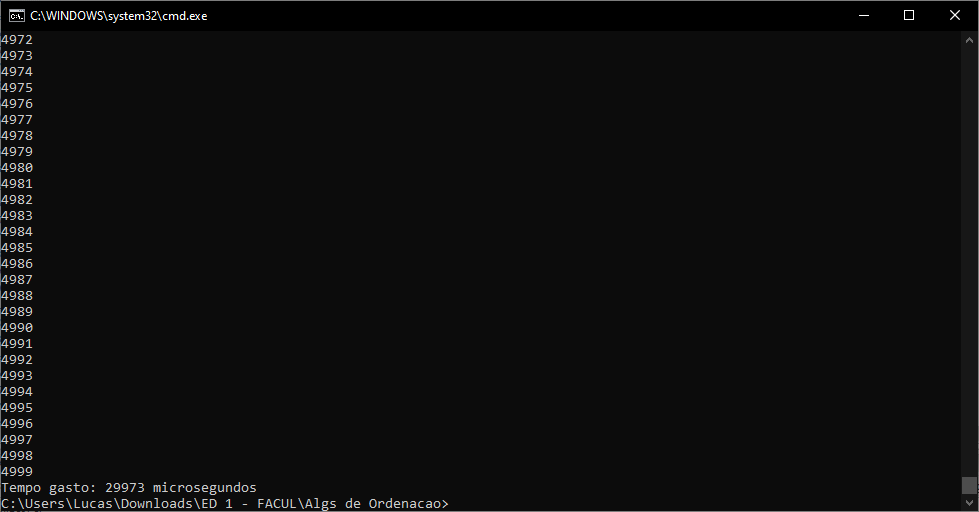
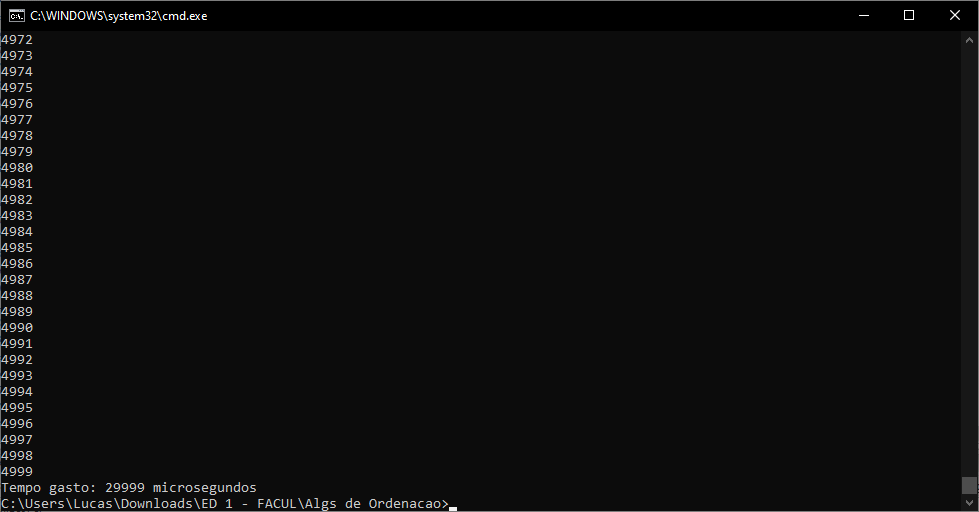
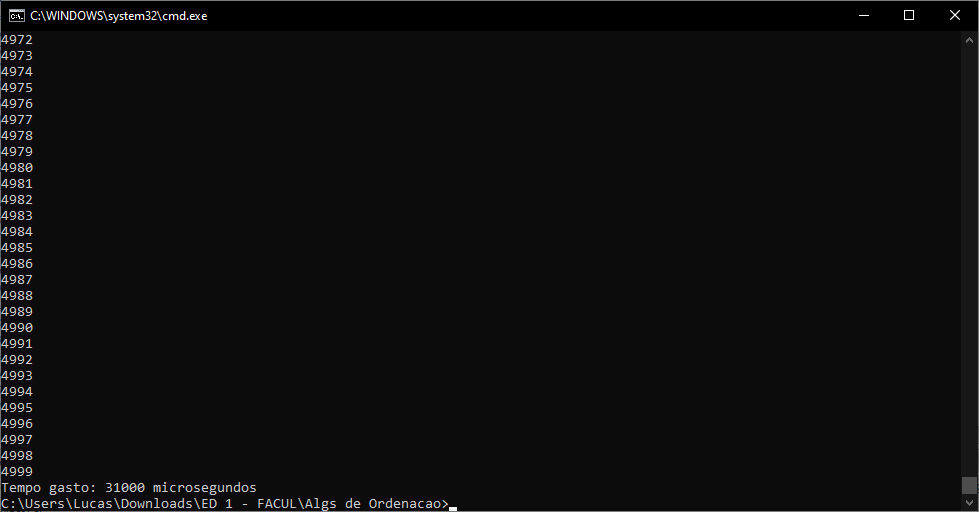
#### **desordenado25k**

1. 
2. 
3. 
4. 
5. 

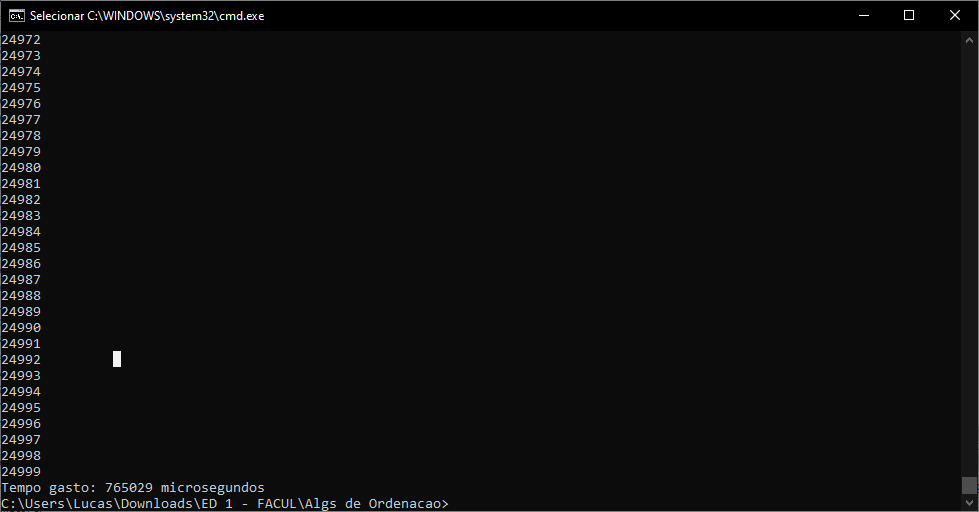
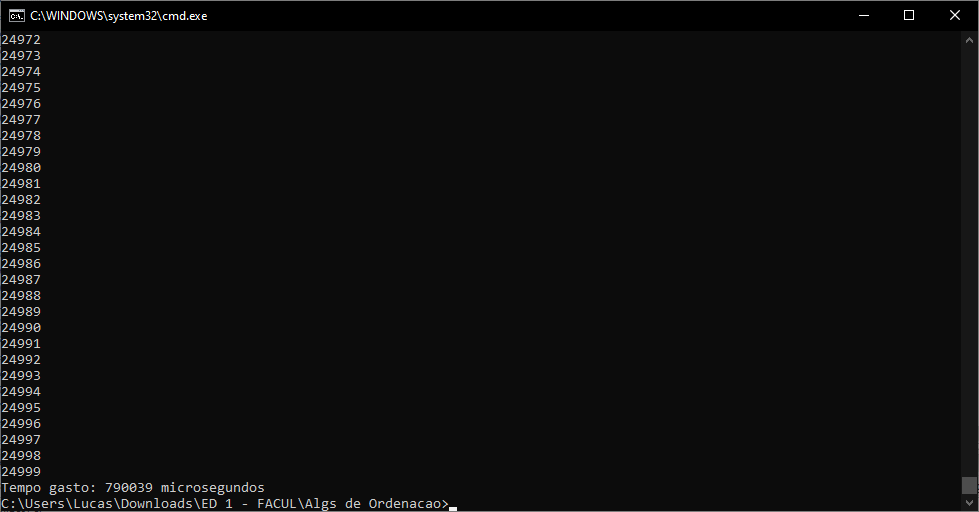
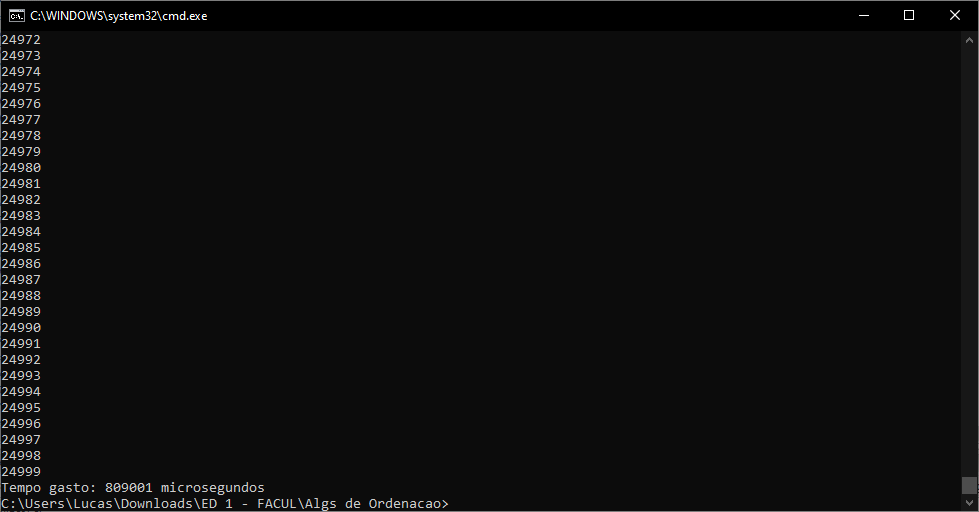
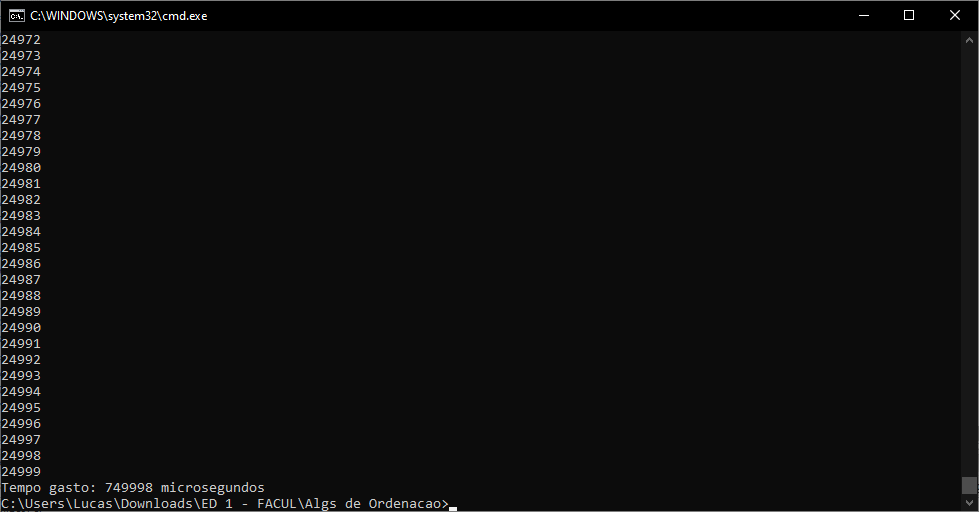
#### **desordenado125k**

1. 
2. 
3. 
4. 
5. 

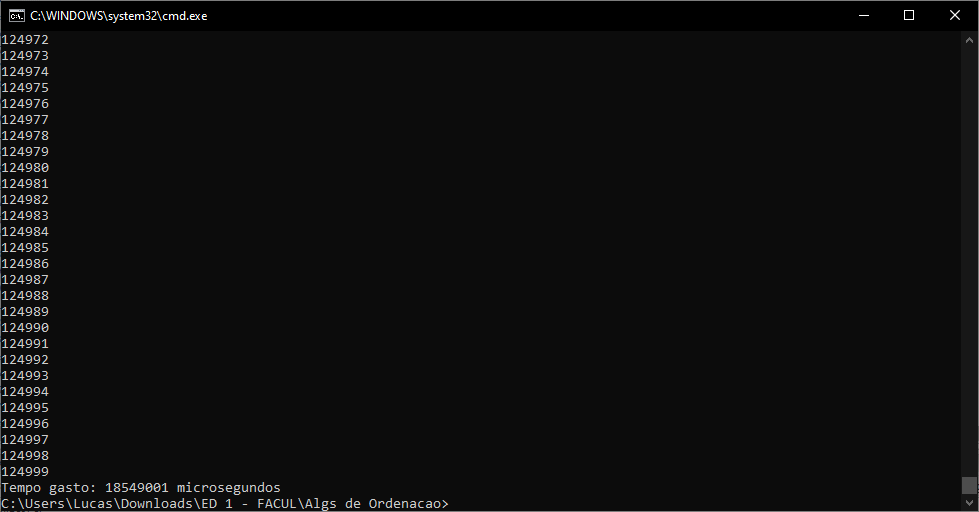
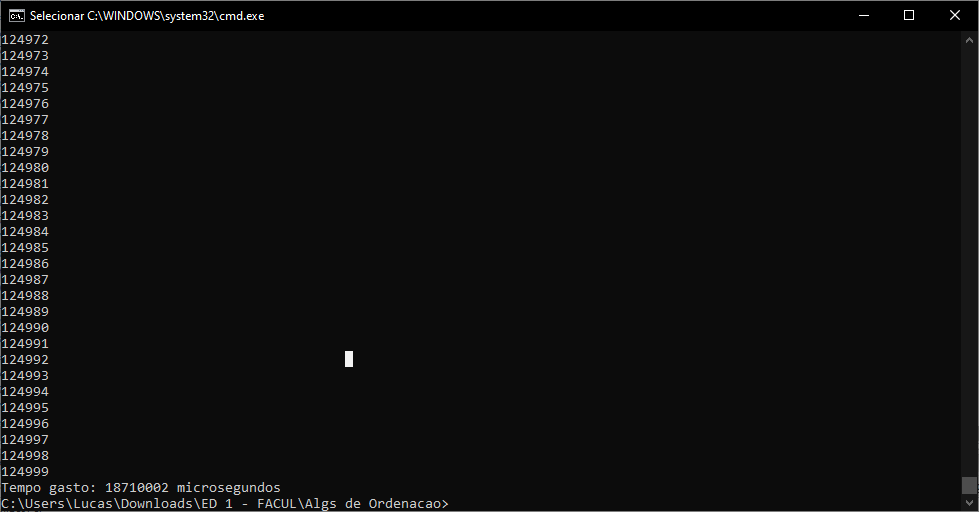
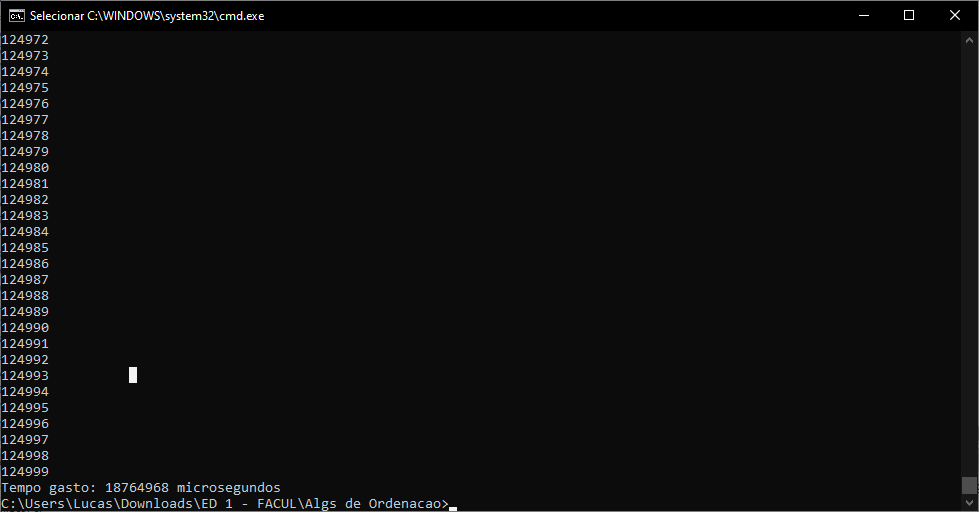
#### **ordenado5k**

1. 
2. 
3. 
4. 
5. 

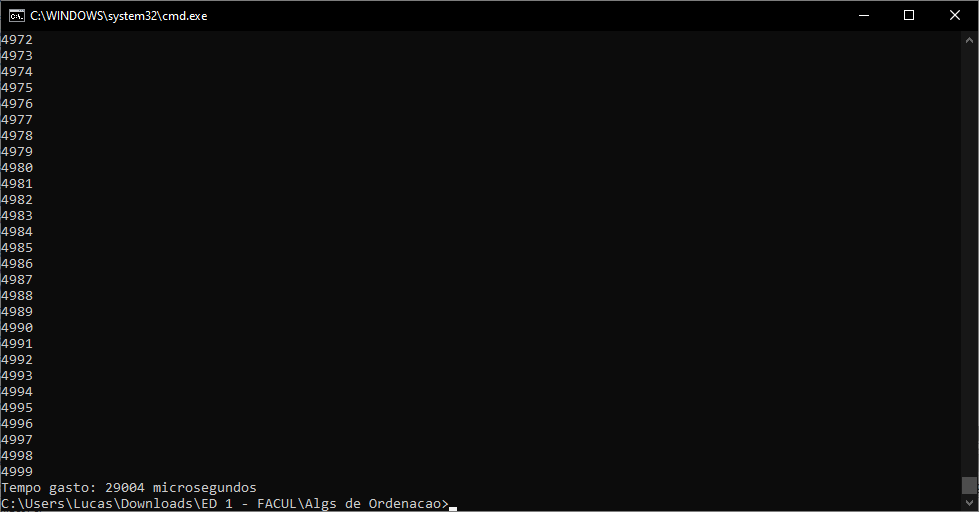
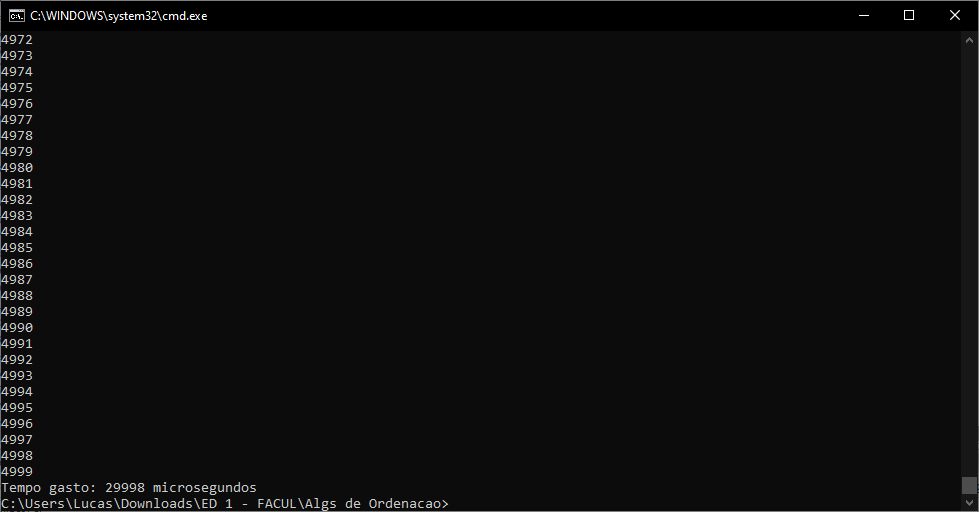
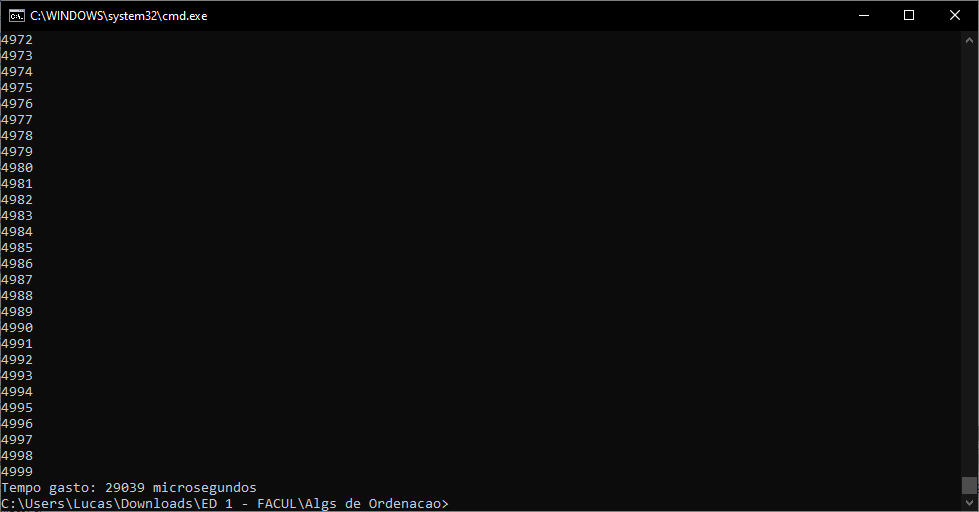
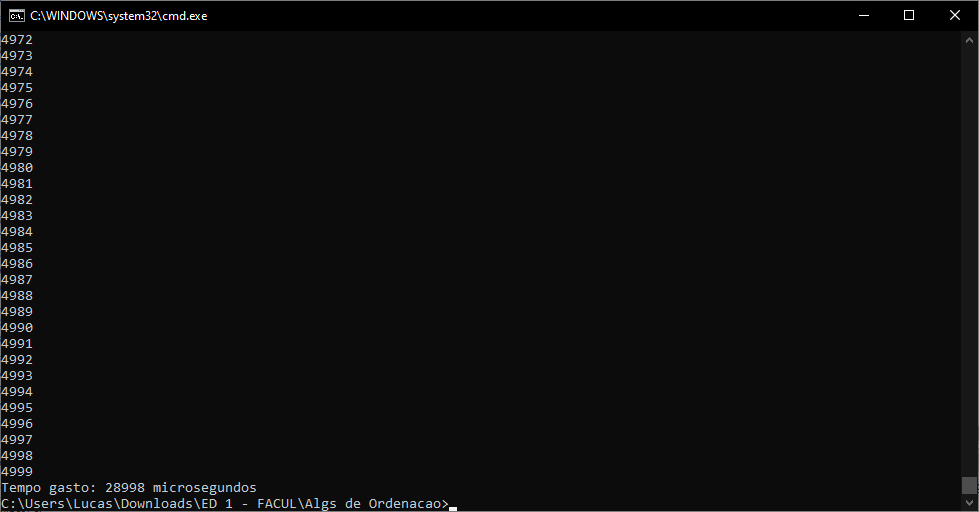
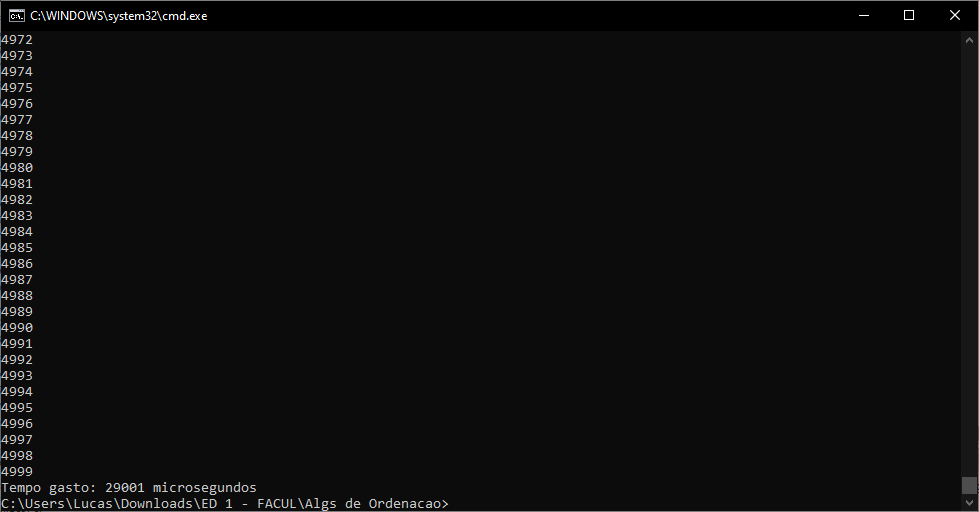
#### **ordenado25k**

1. 
2. 
3. 
4. 
5. 

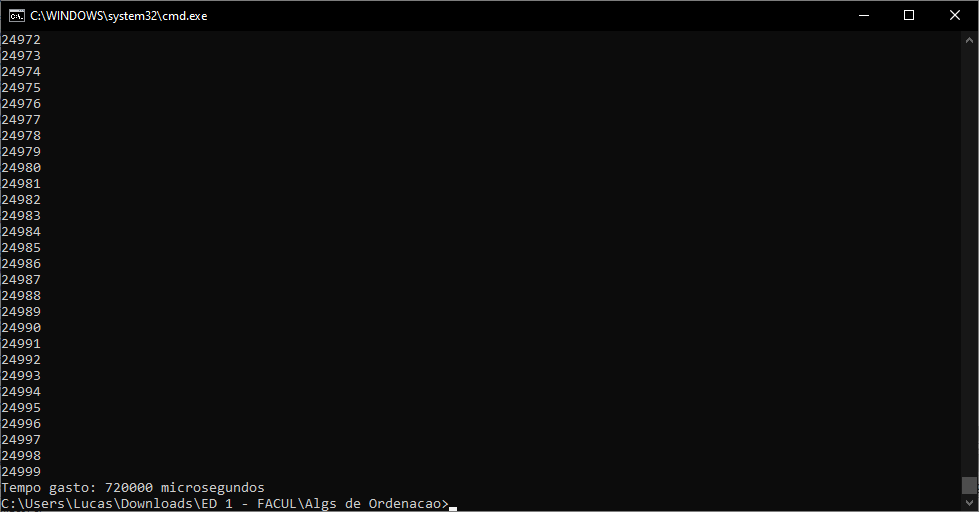
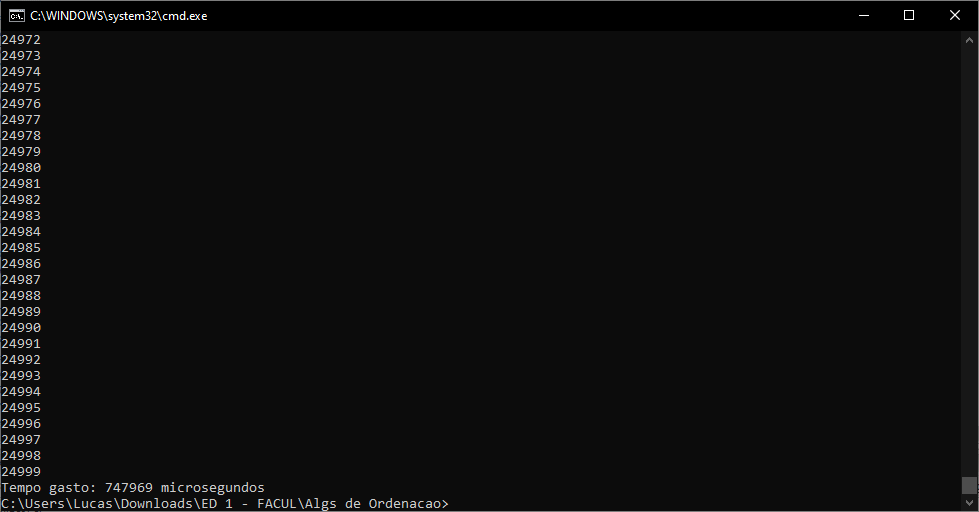
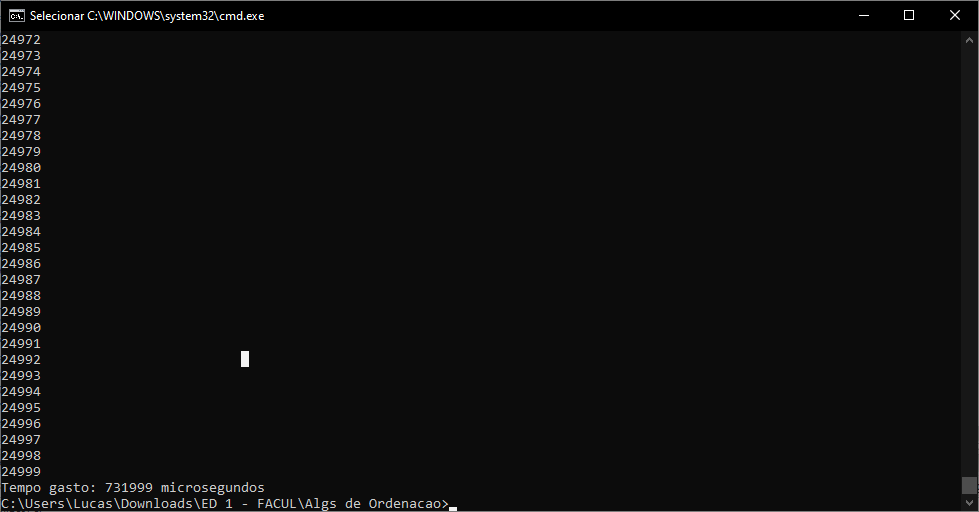
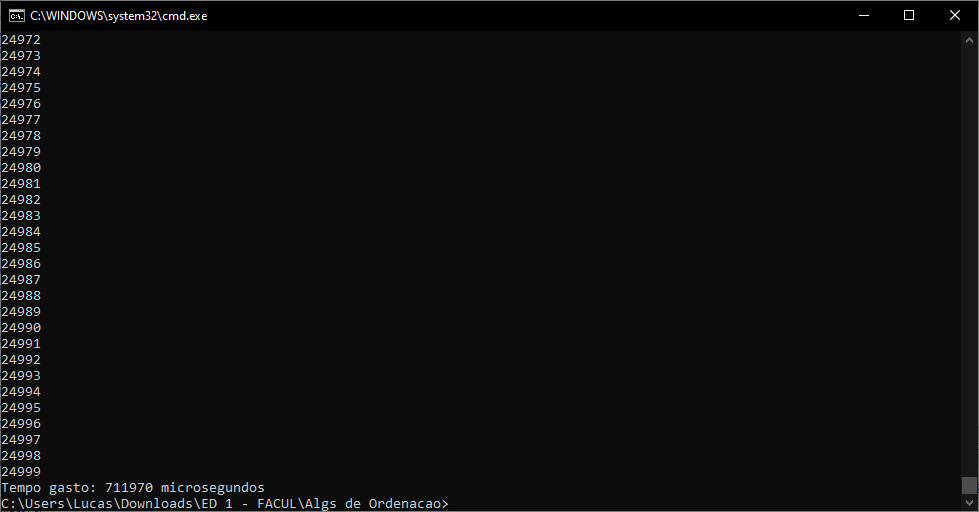
#### **ordenado125k**

1. 
2. 
3. 
4. 
5. 

#### **ordenadodesc5k**

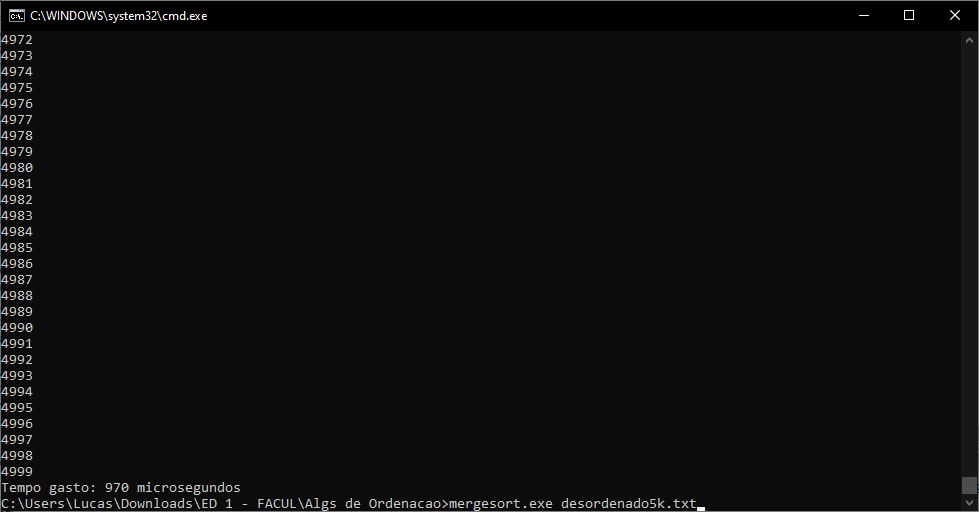
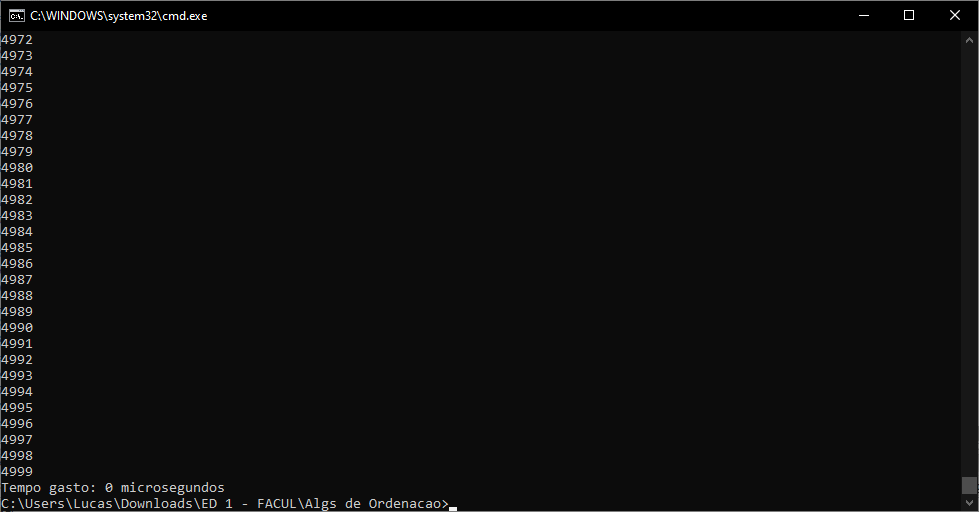
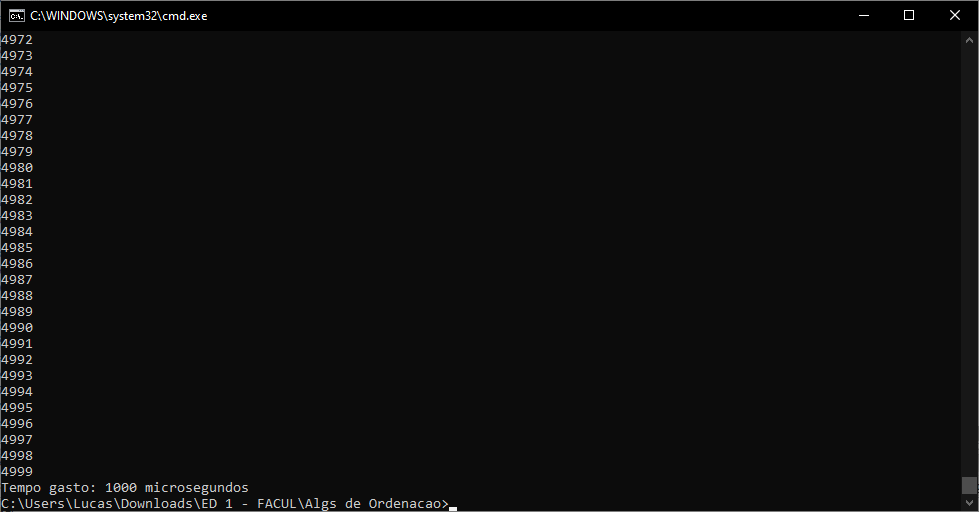
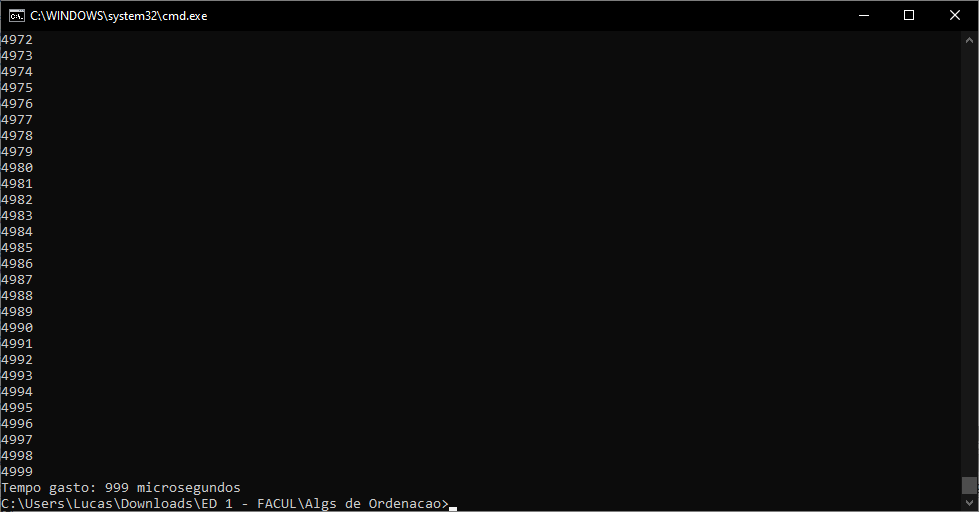
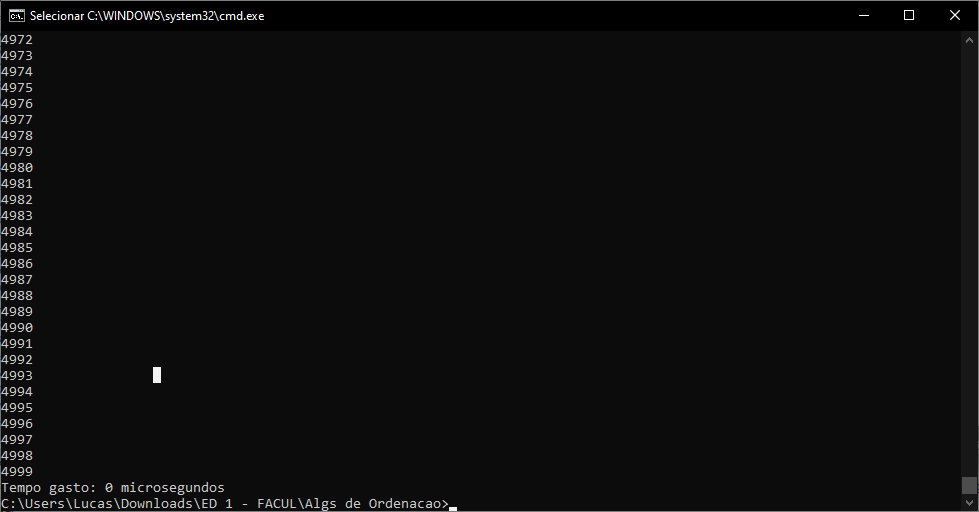
1. 
2. 
3. 
4. 
5. 

#### **ordenadodesc25k**

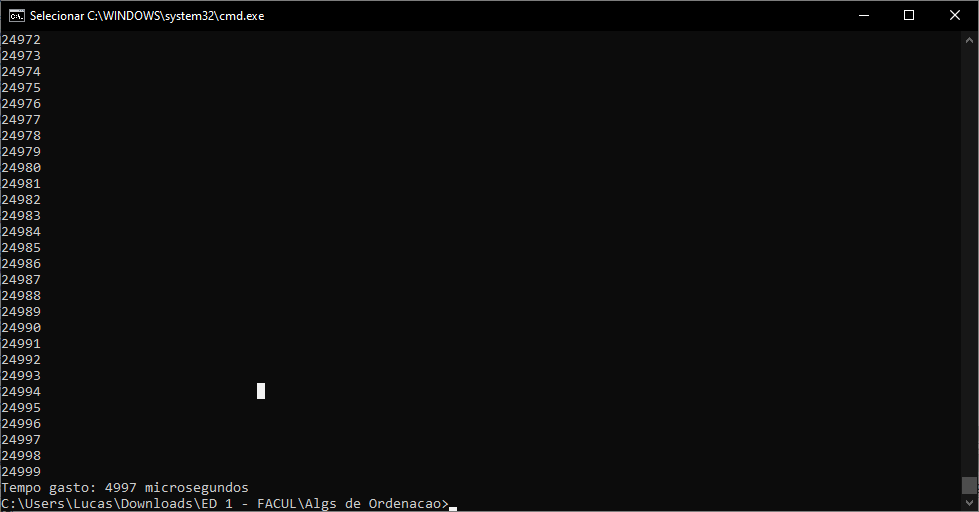
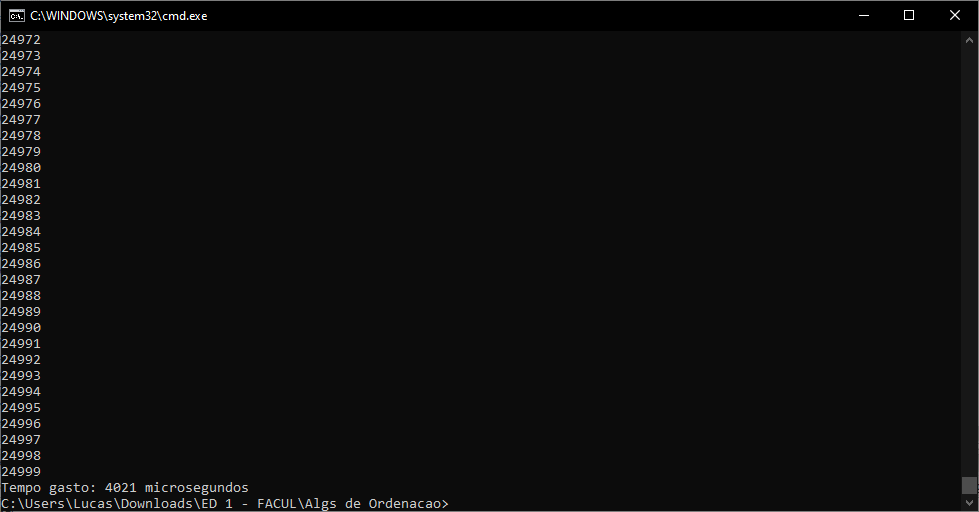
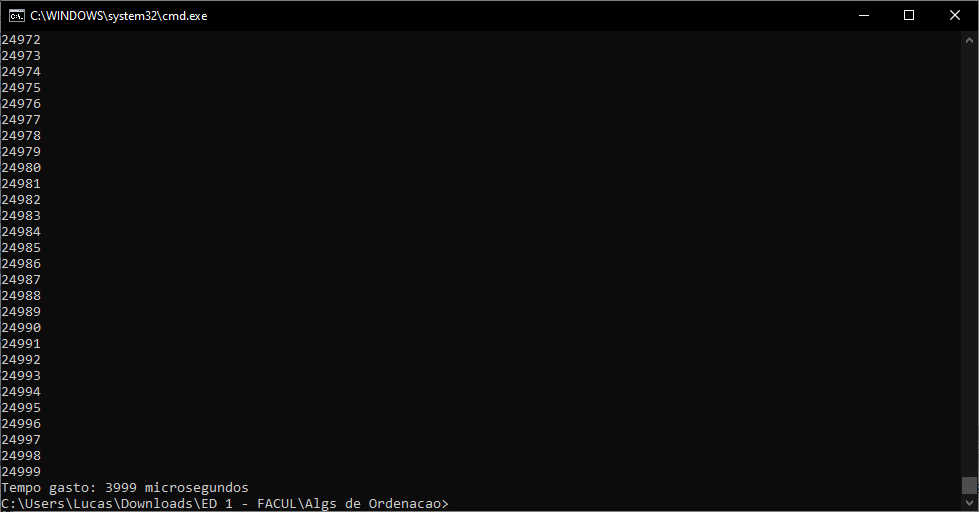
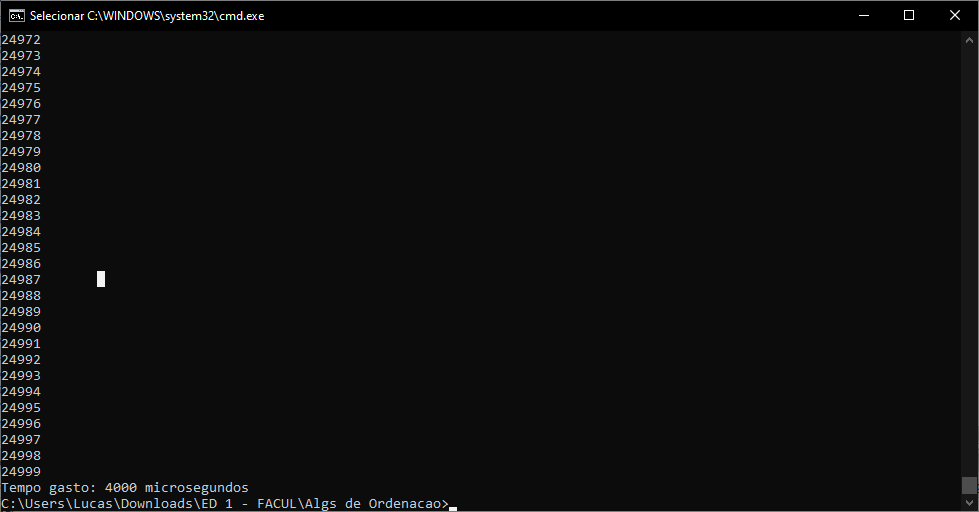
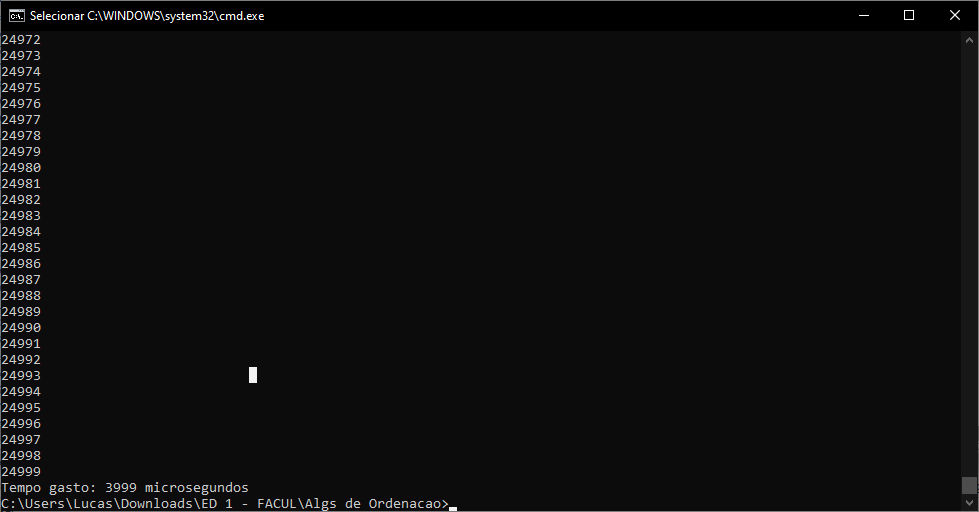
1. 
2. 
3. 
4. 
5. 

### ***MergeSort***

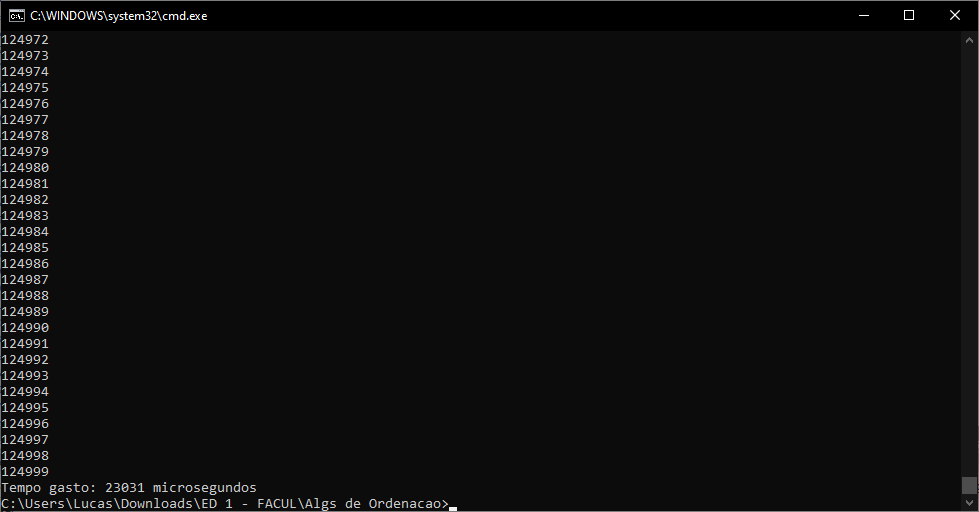
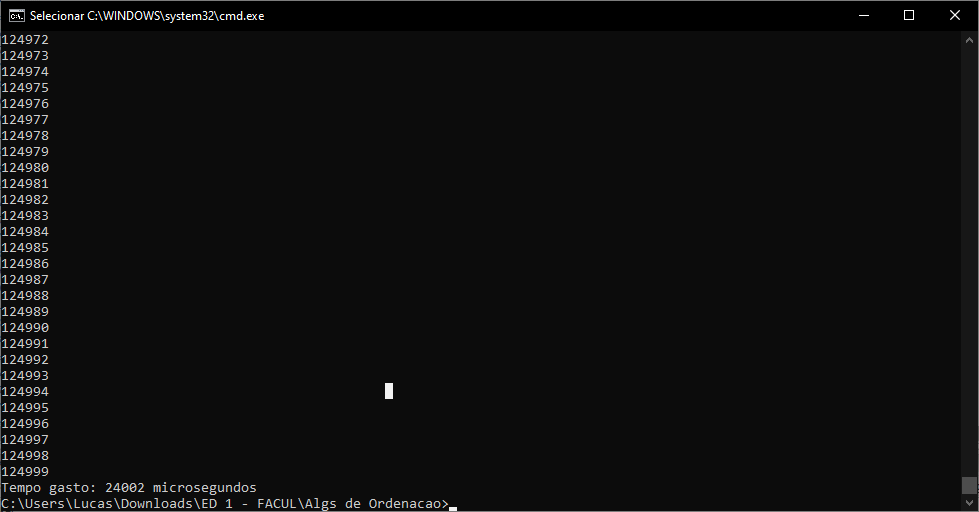
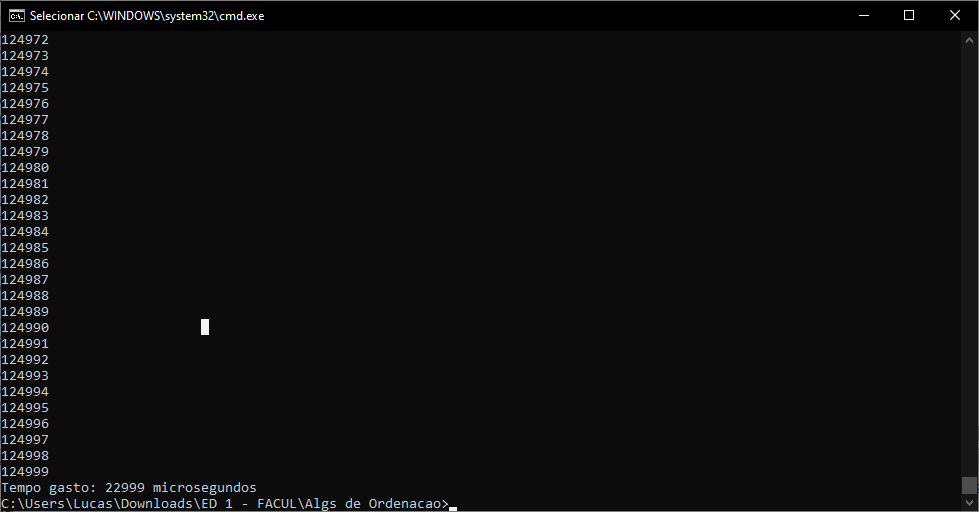
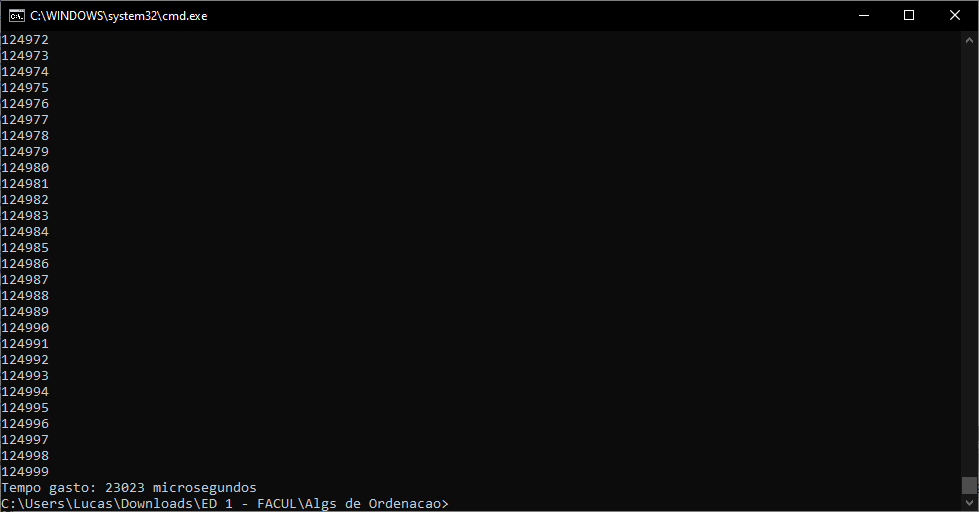
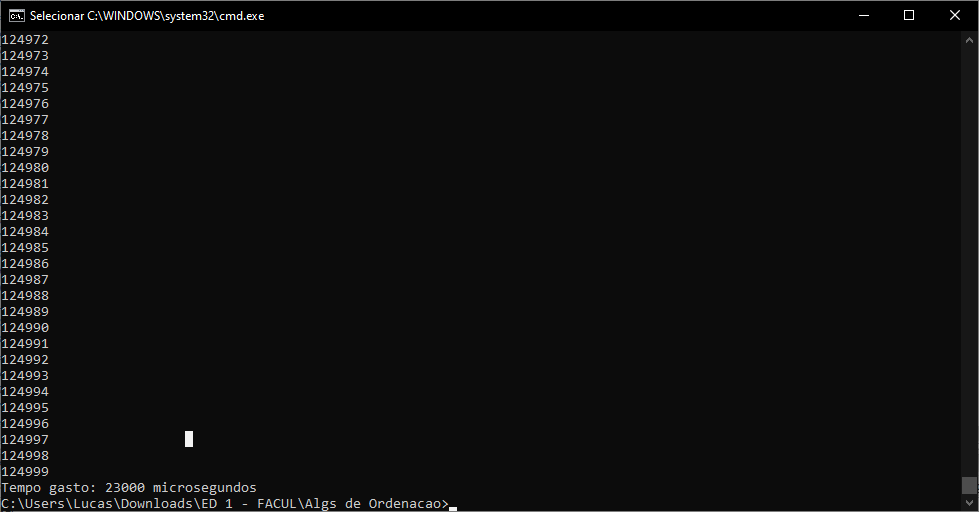
#### **desordenado5k**

1. 
2. 
3. 
4. 
5. 

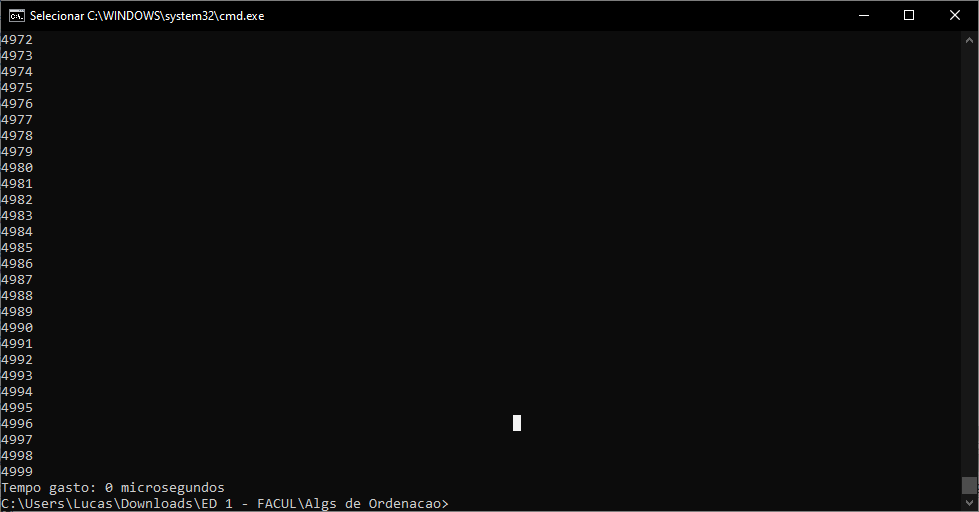
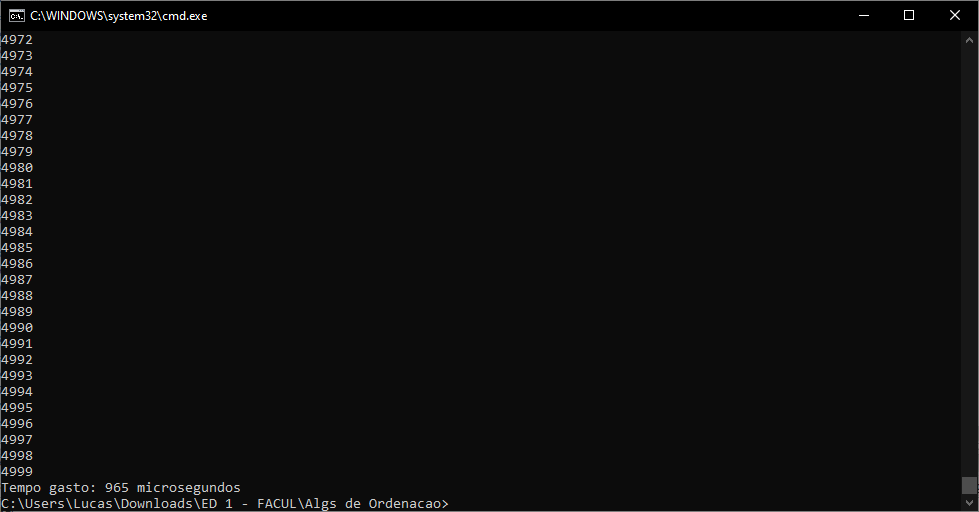
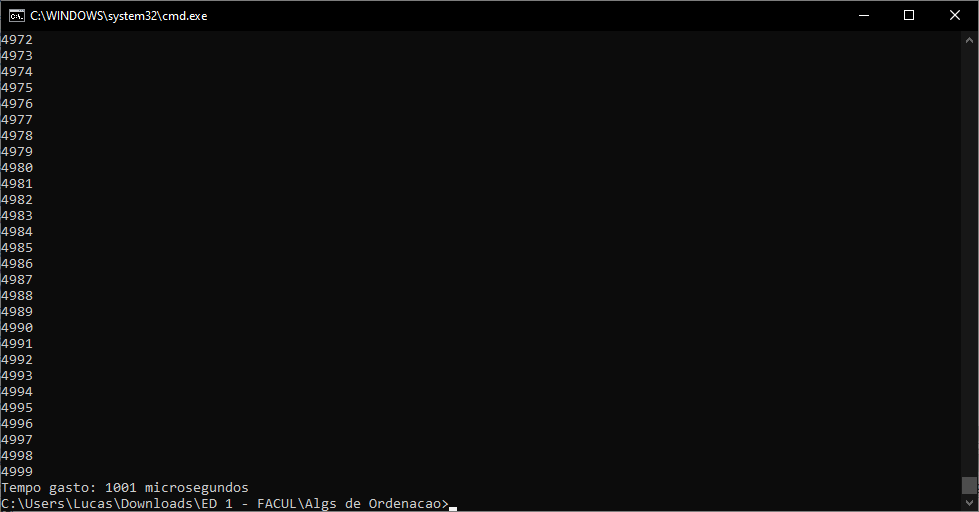
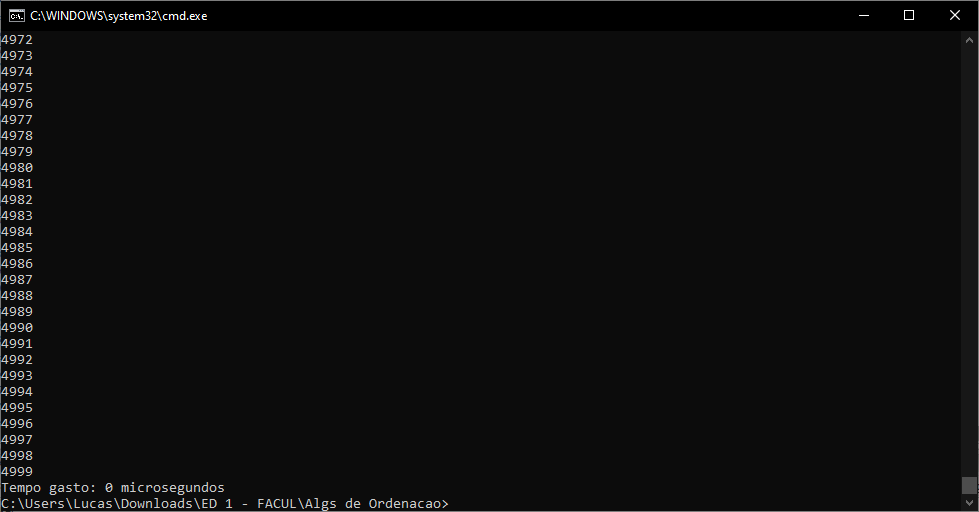
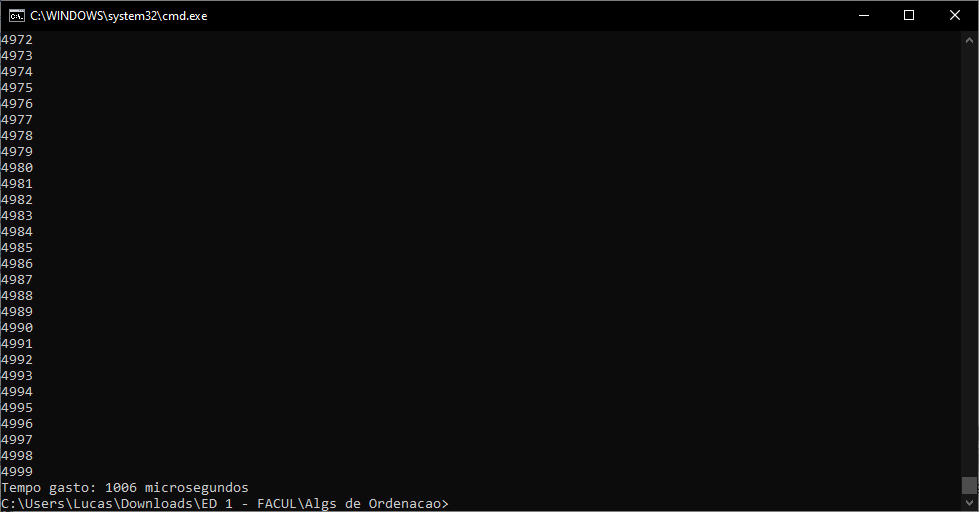
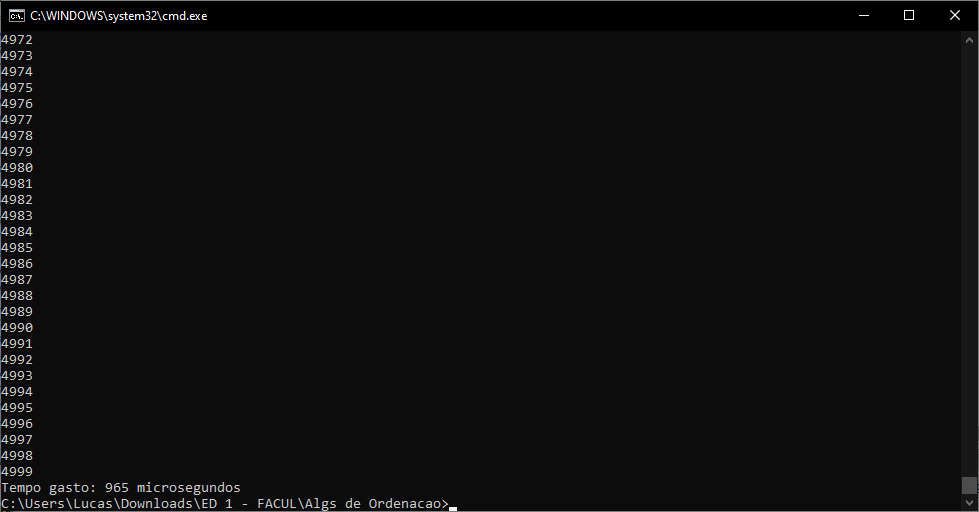
#### **desordenado25k**

1. 
2. 
3. 
4. 
5. 

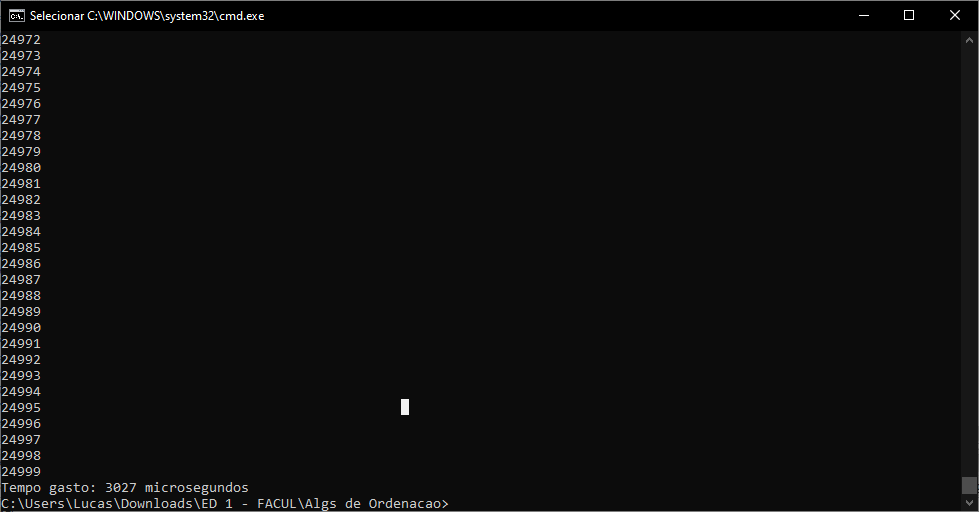
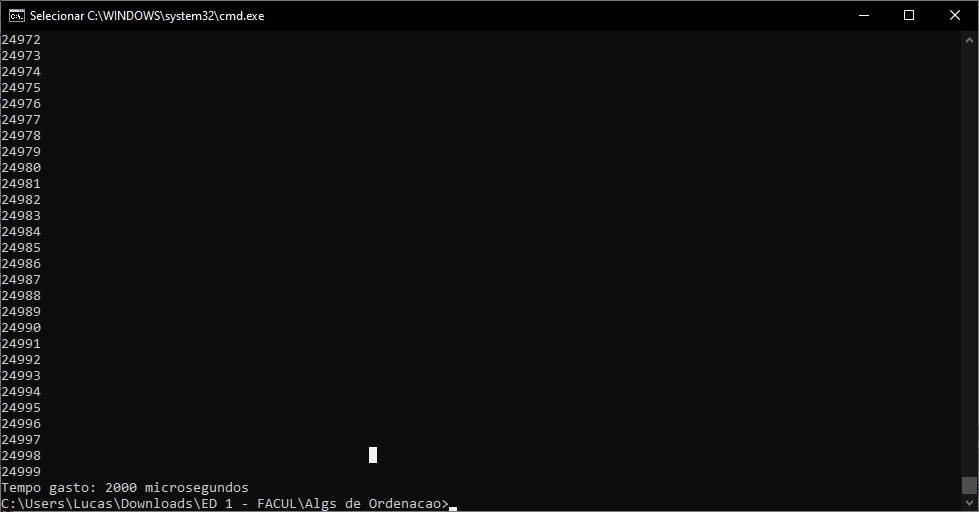
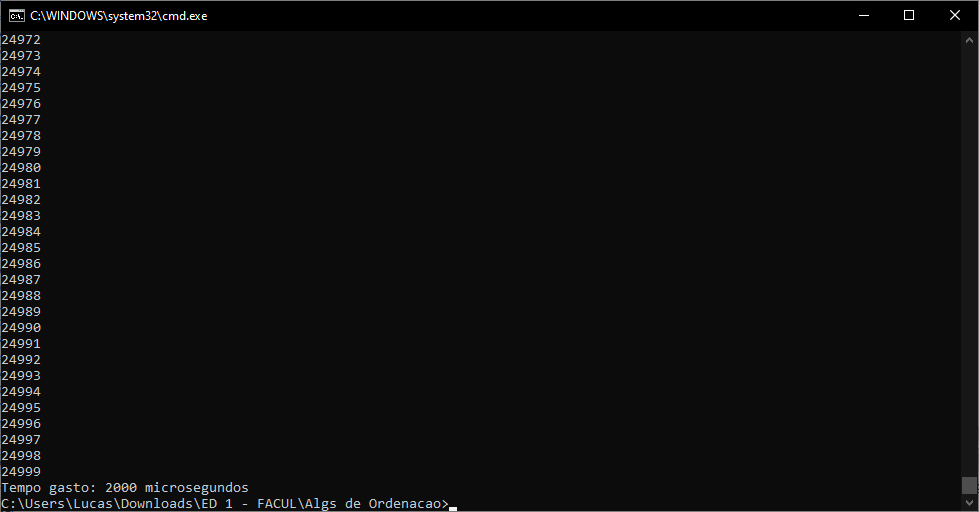
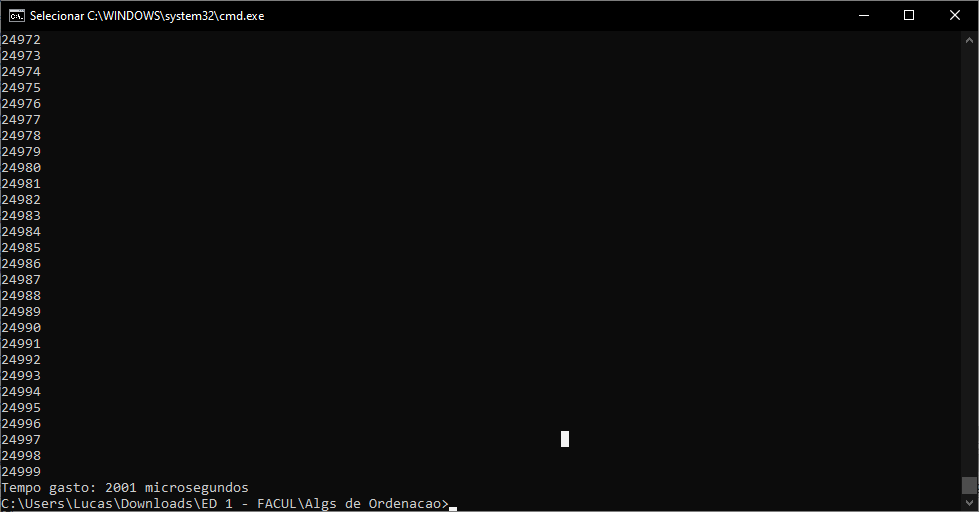
#### **desordenado125k**

1. 
2. 
3. 
4. 
5. 

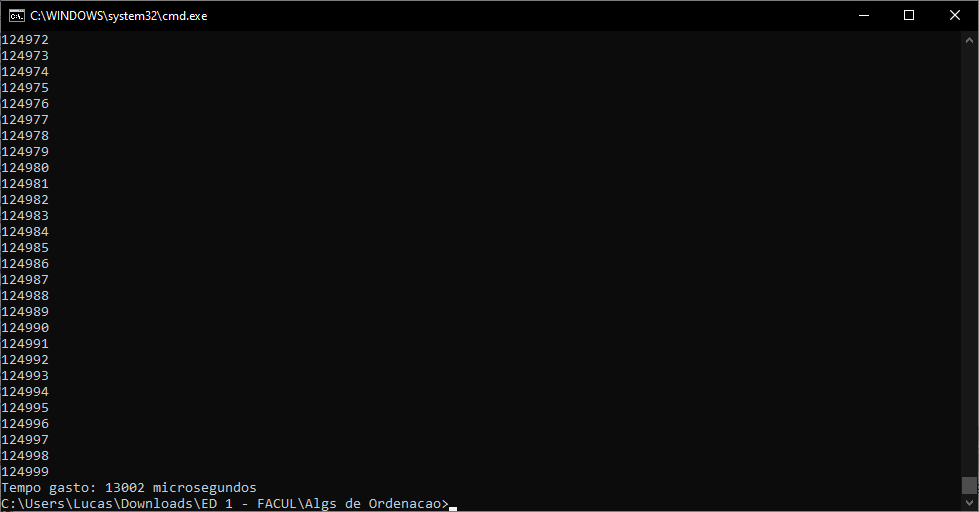
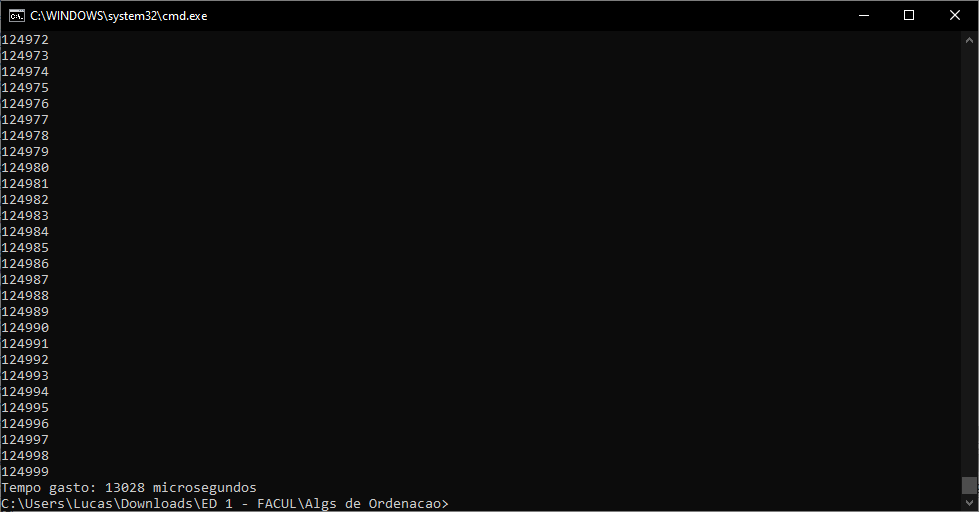
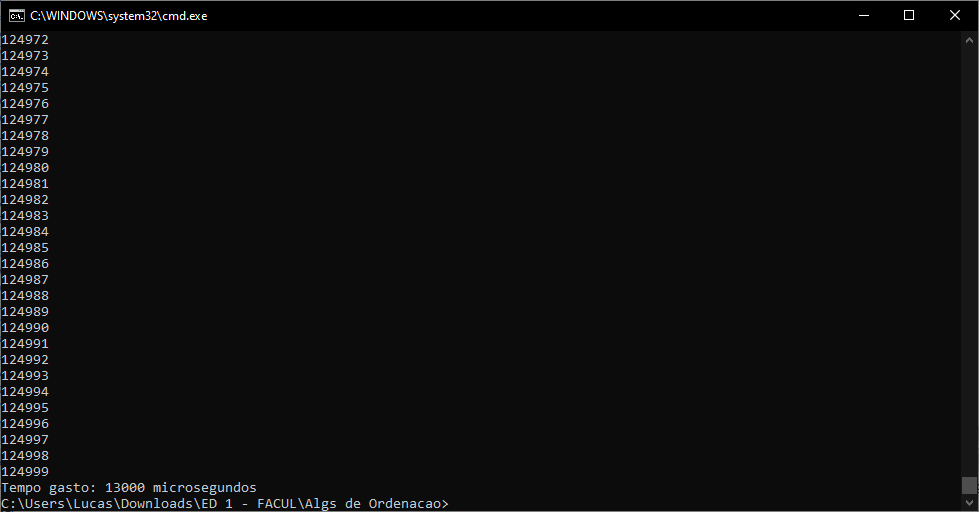
#### **ordenado5k**

1. 
2. 
3. 
4. 
5. 
6. 

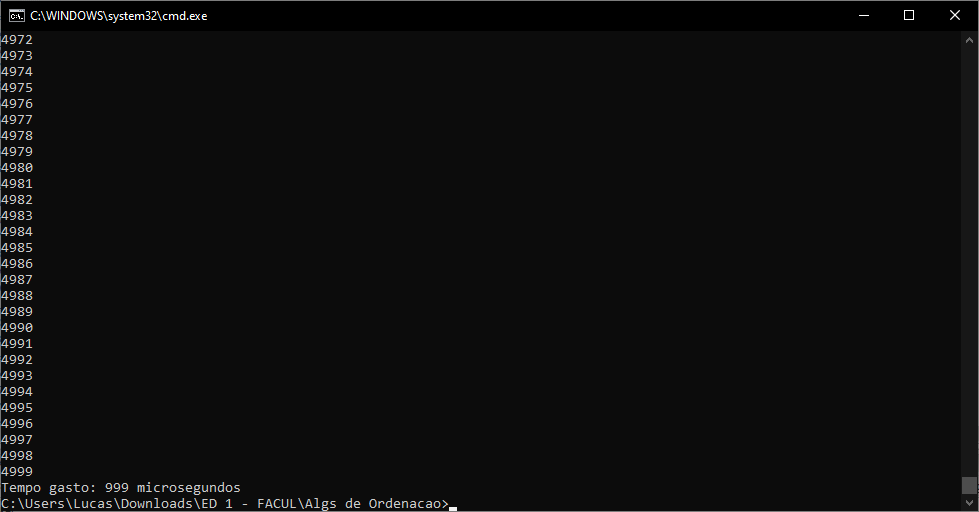
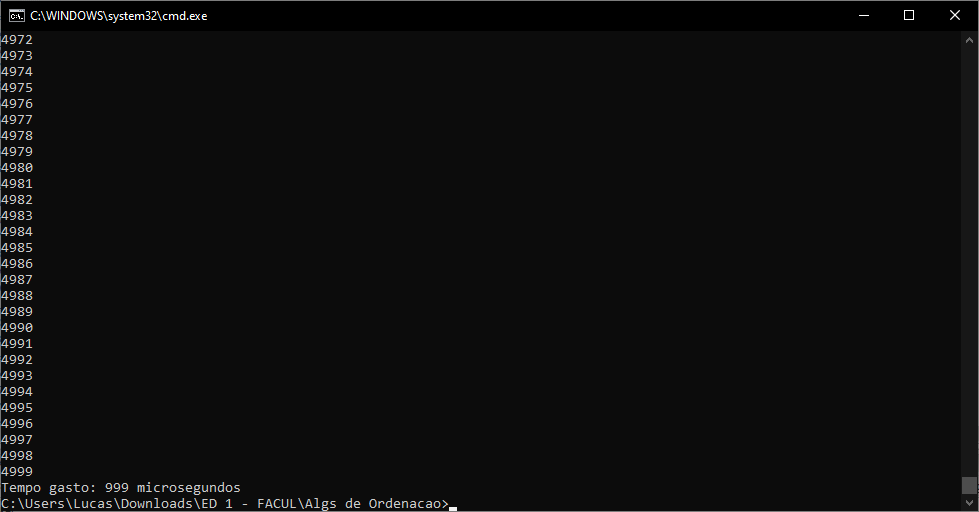
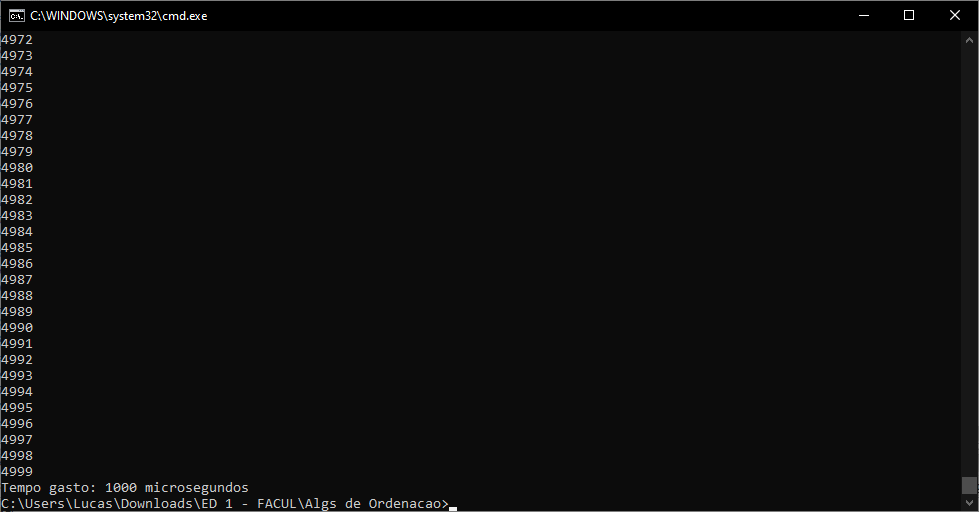
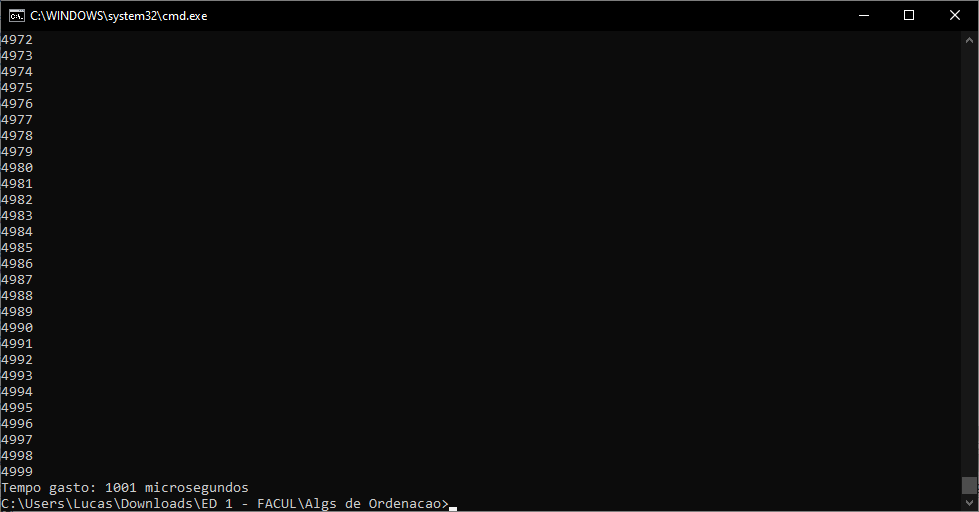
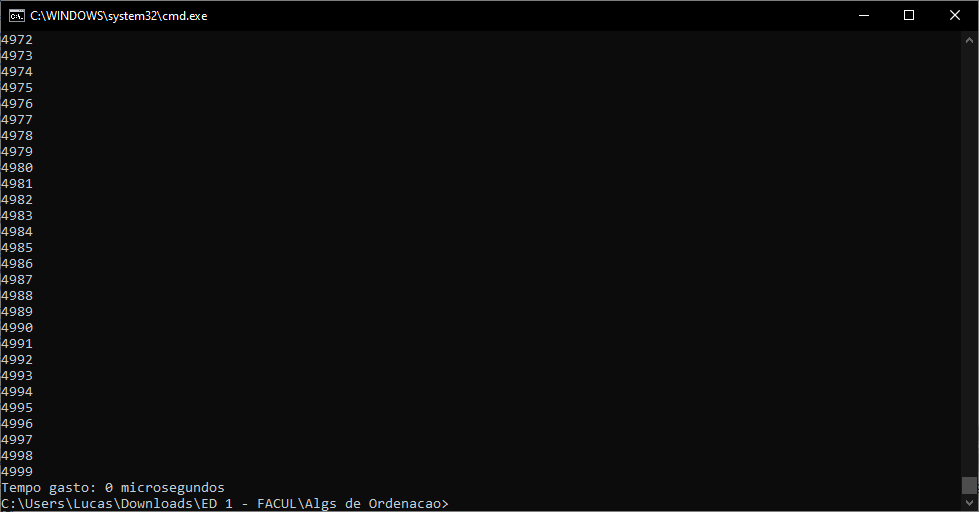
#### **ordenado25k**

1. 
2. 
3. 
4. 
5. 

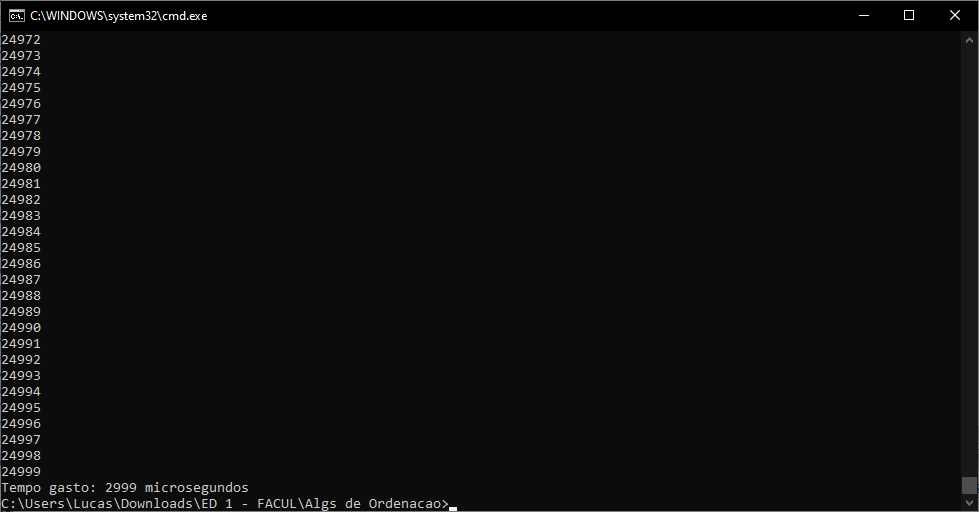
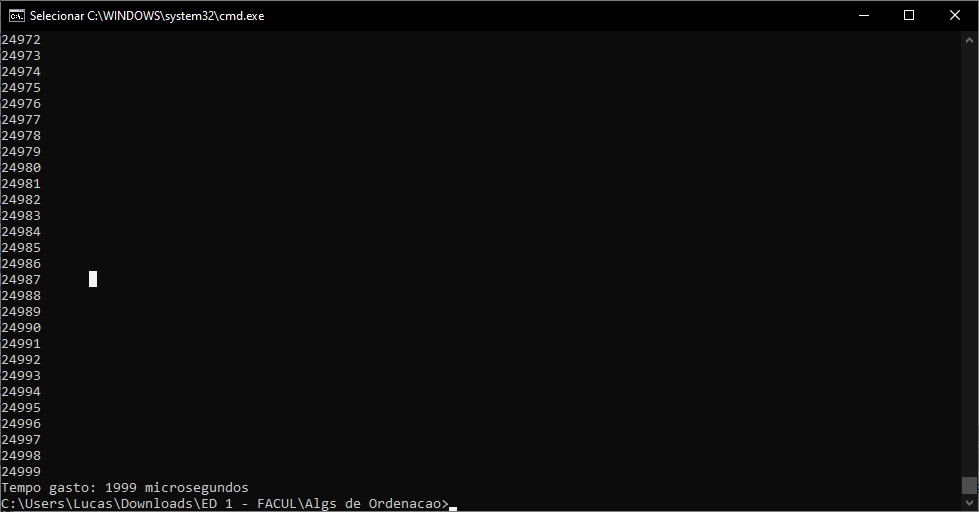
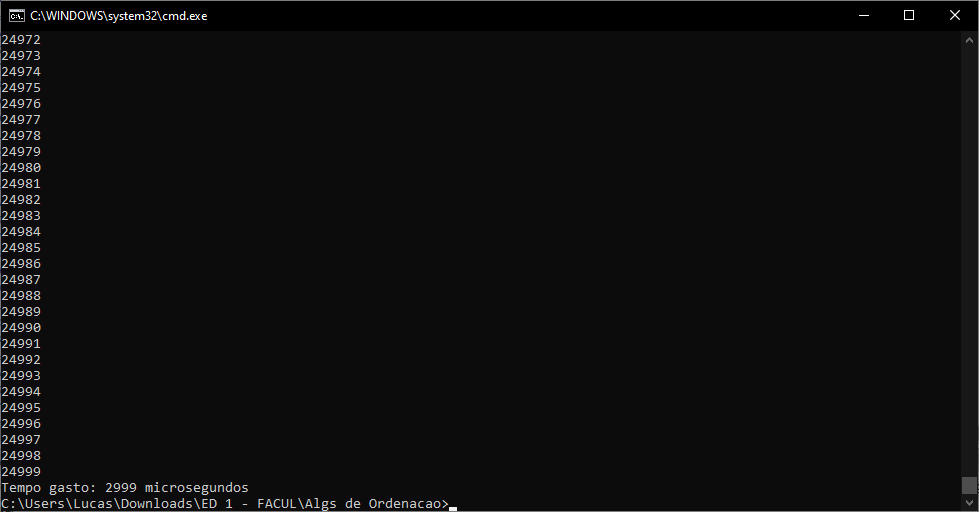
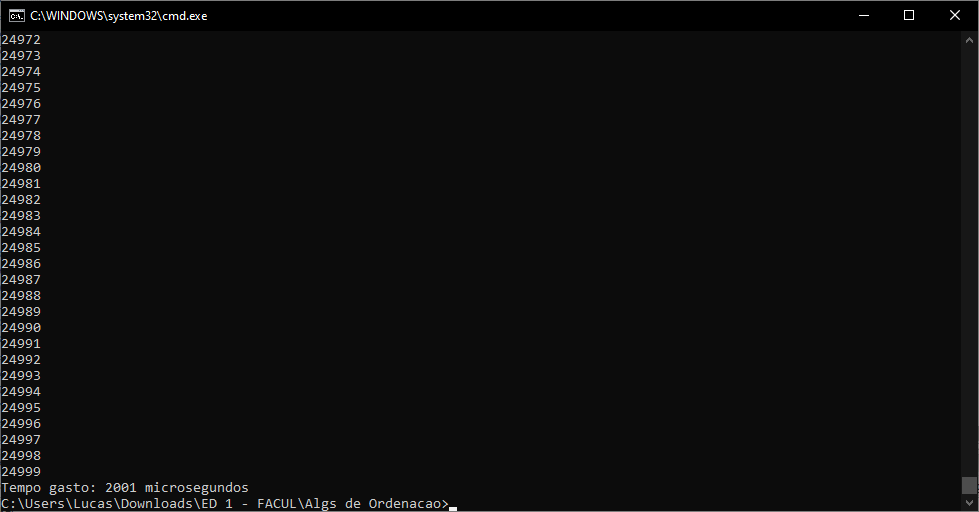
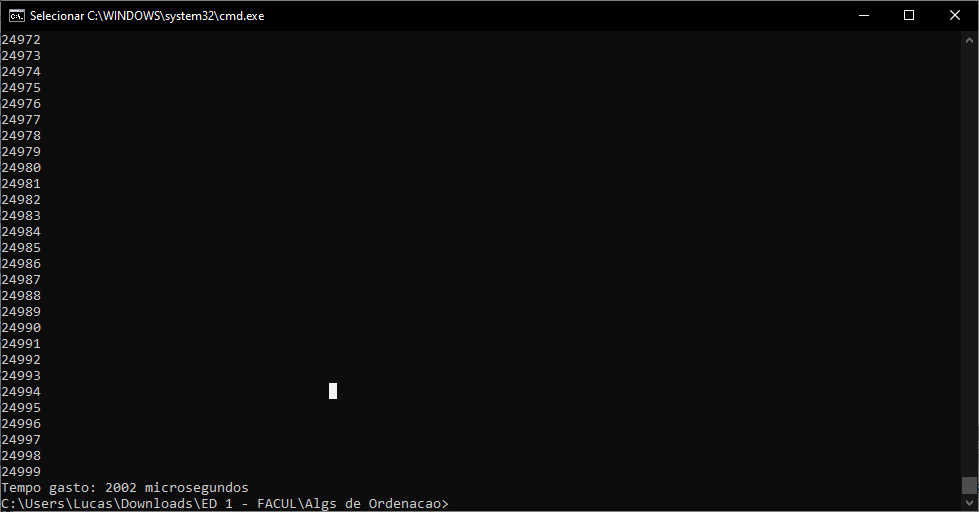
#### **ordenado125k**

1. 
2. 
3. 
4. 
5. 

#### **ordenadodesc5k**

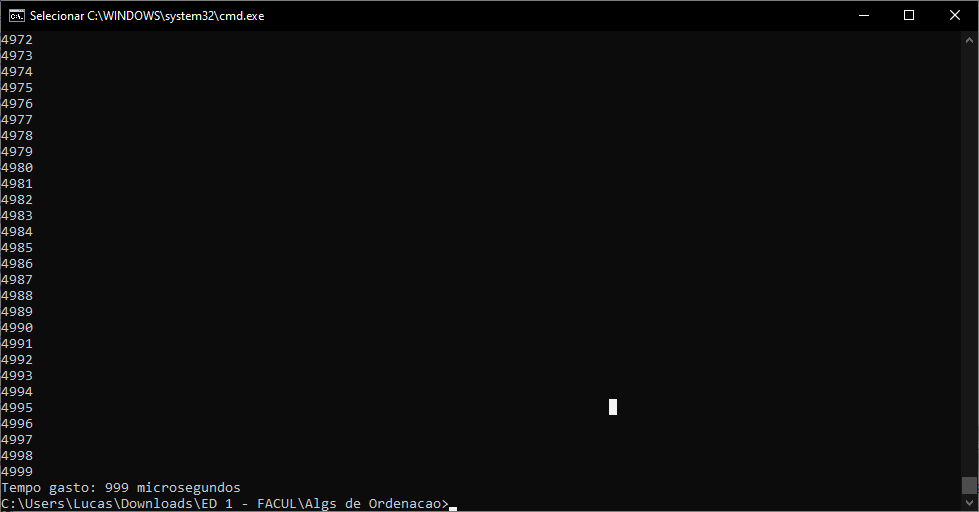
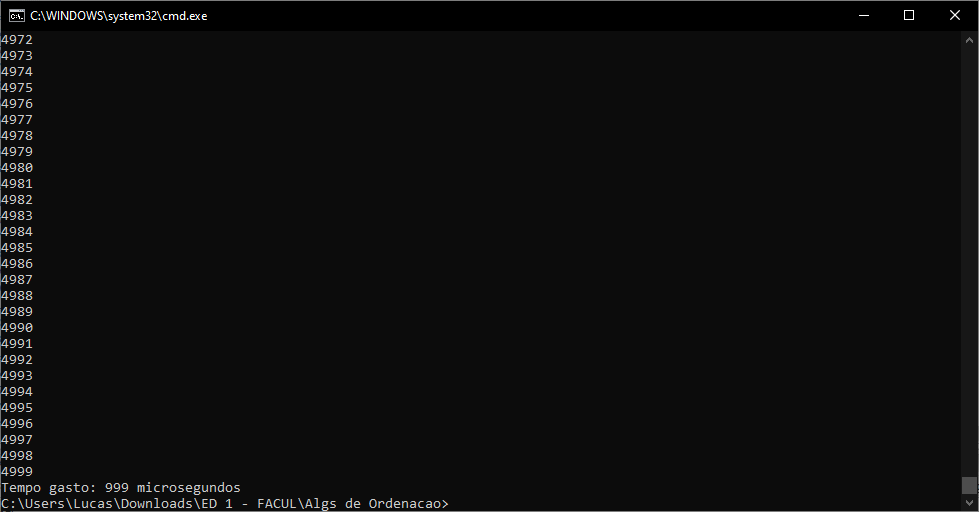
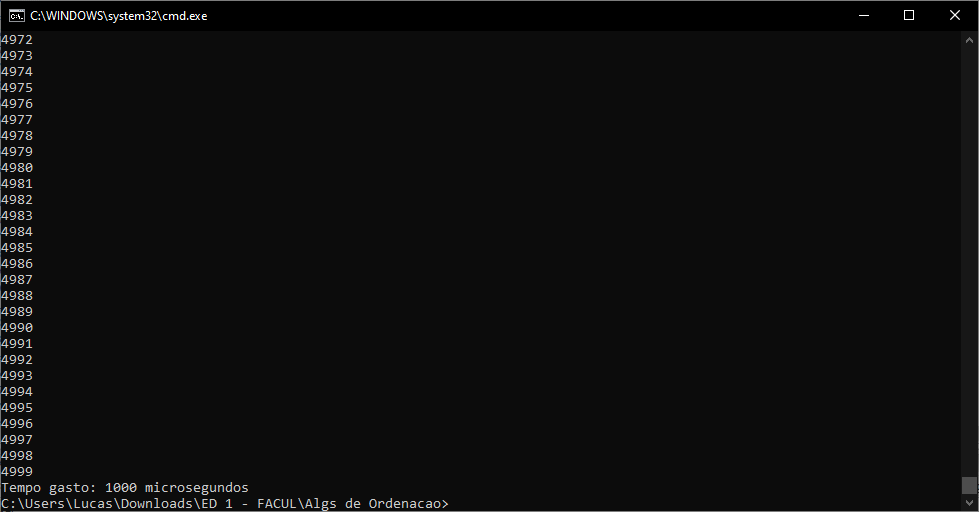
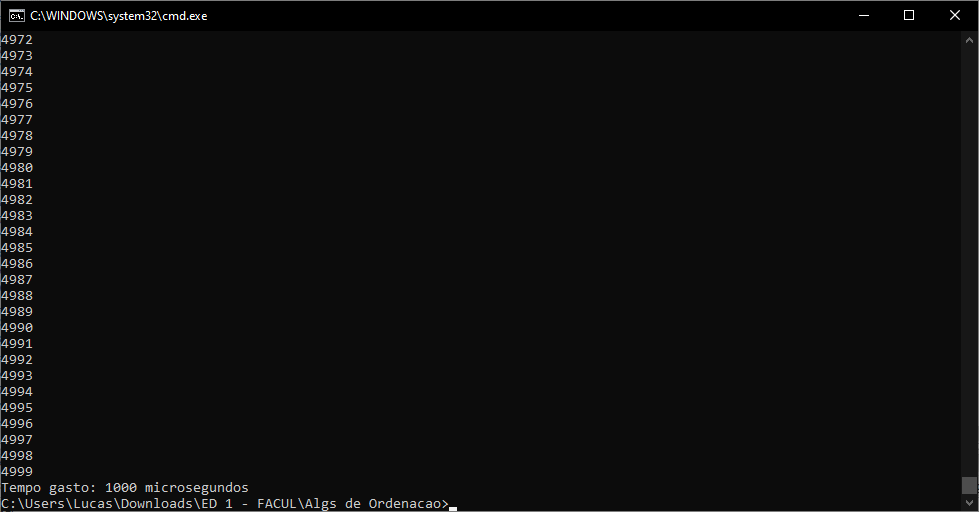
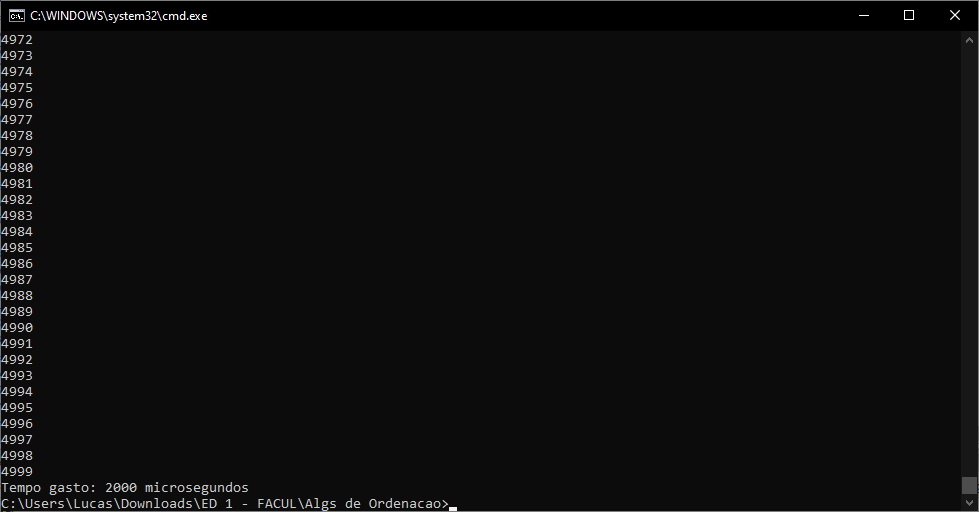
1. 
2. 
3. 
4. 
5. 

#### **ordenadodesc25k**

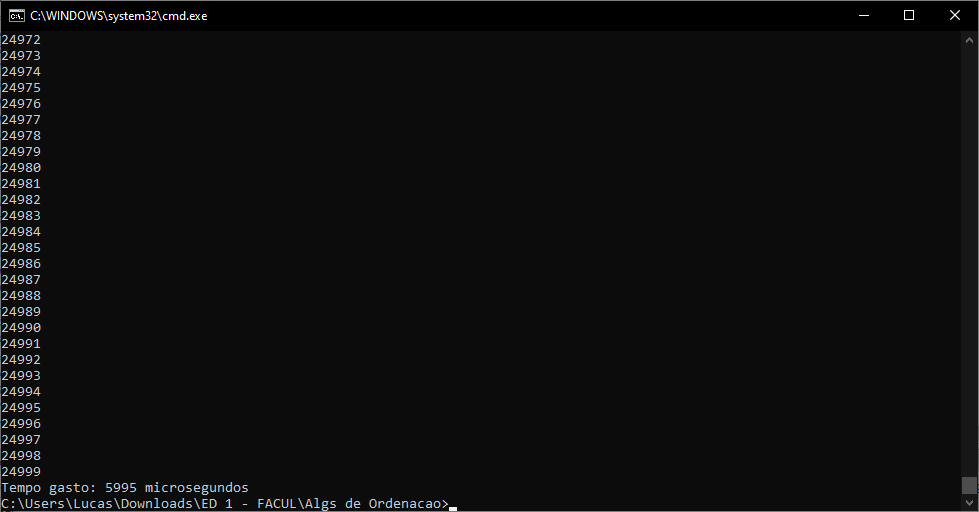
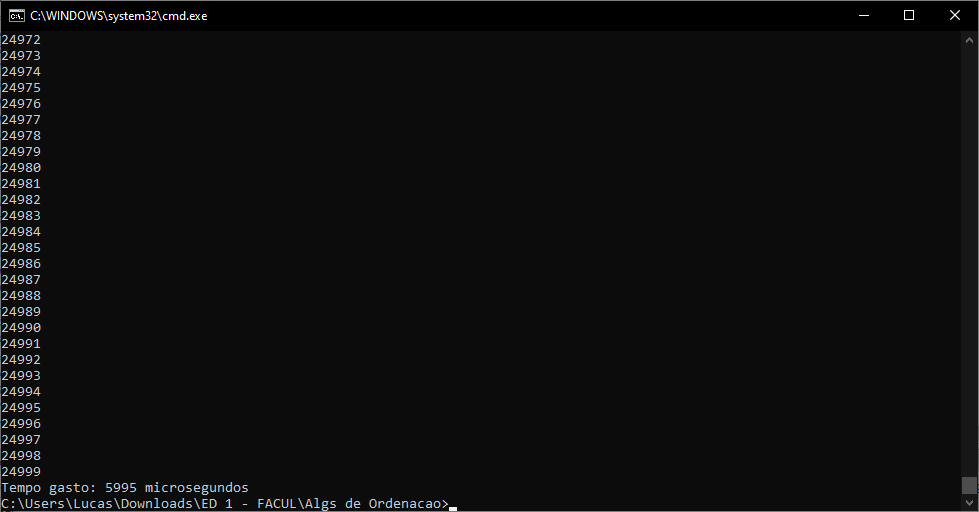
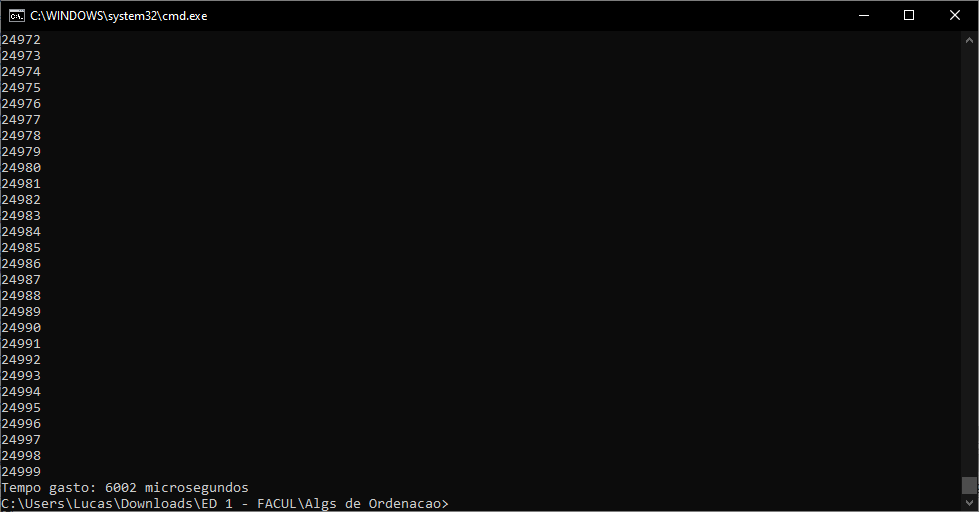
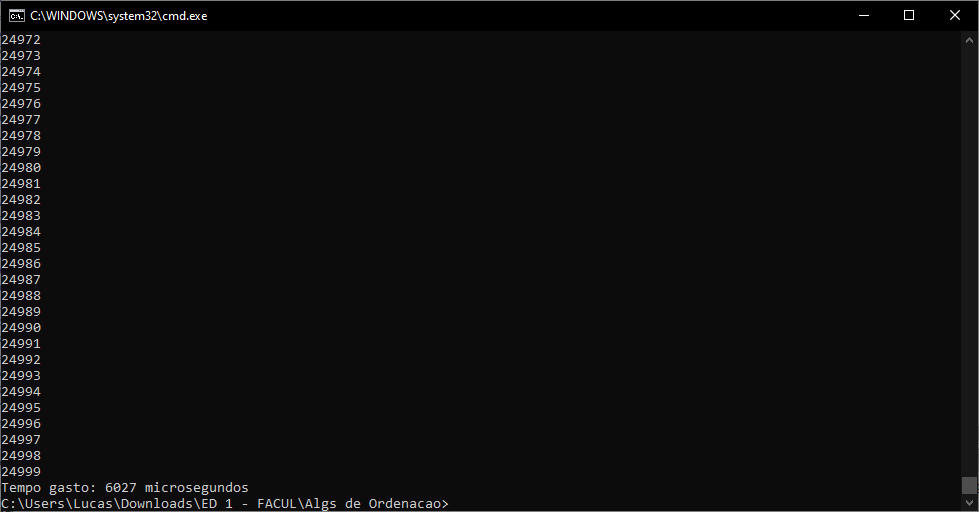
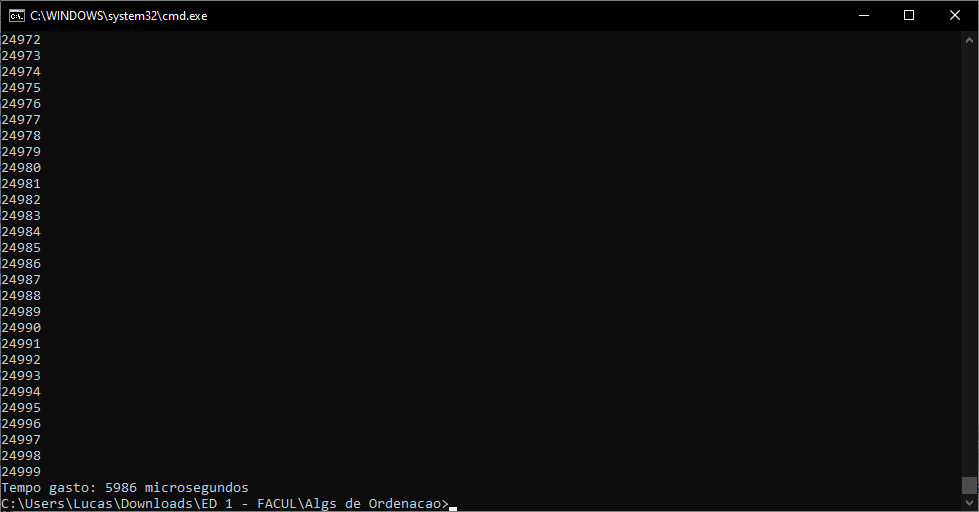
1. 
2. 
3. 
4. 
5. 

### ***HeapSort***

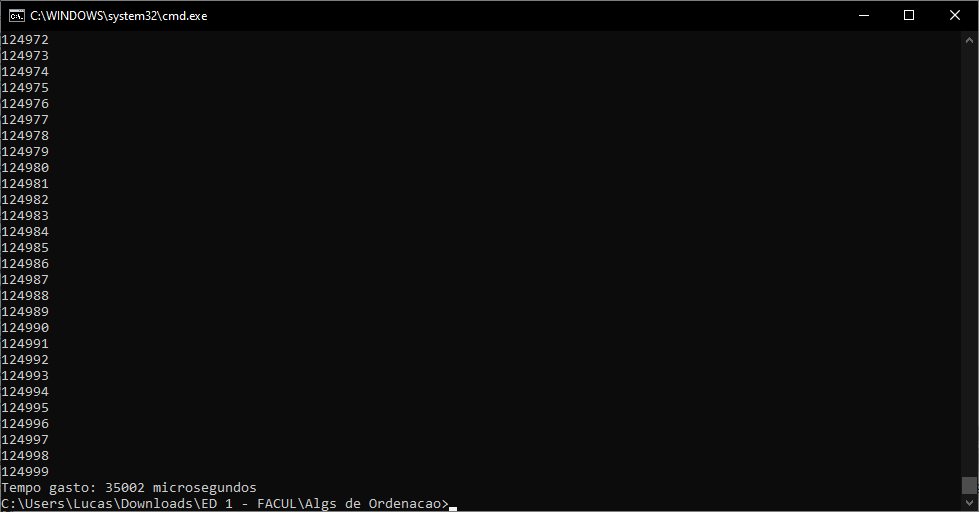
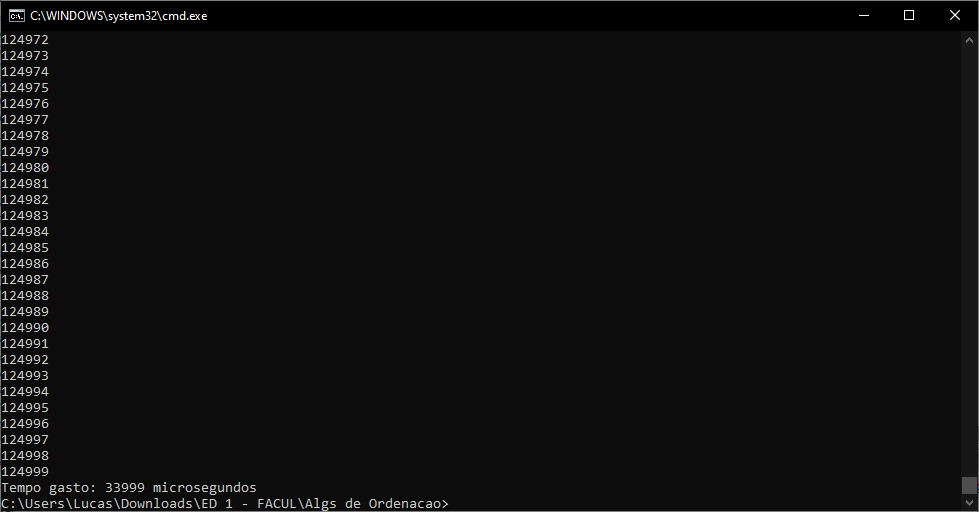
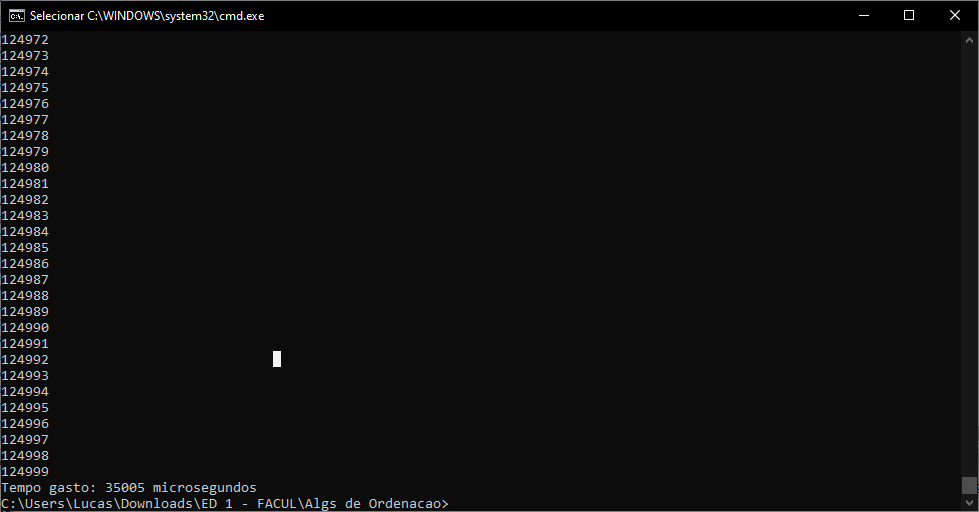
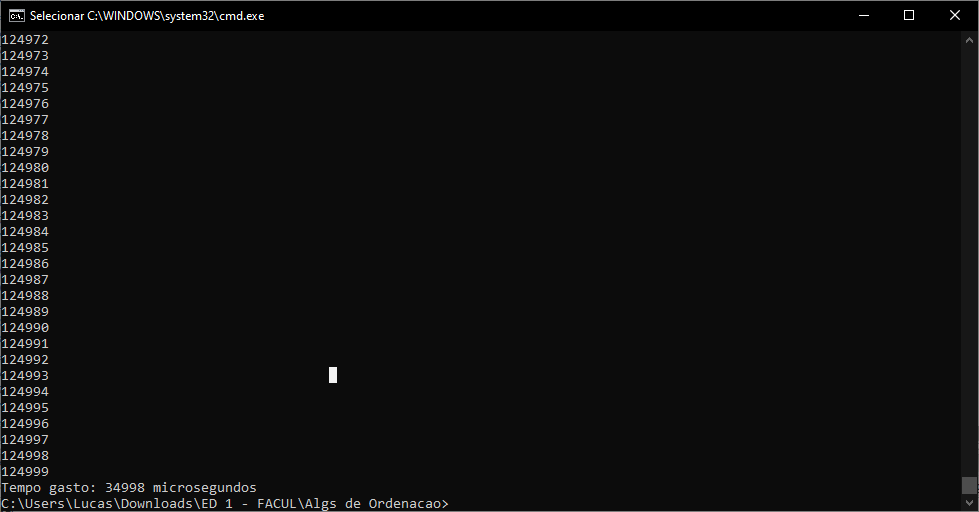
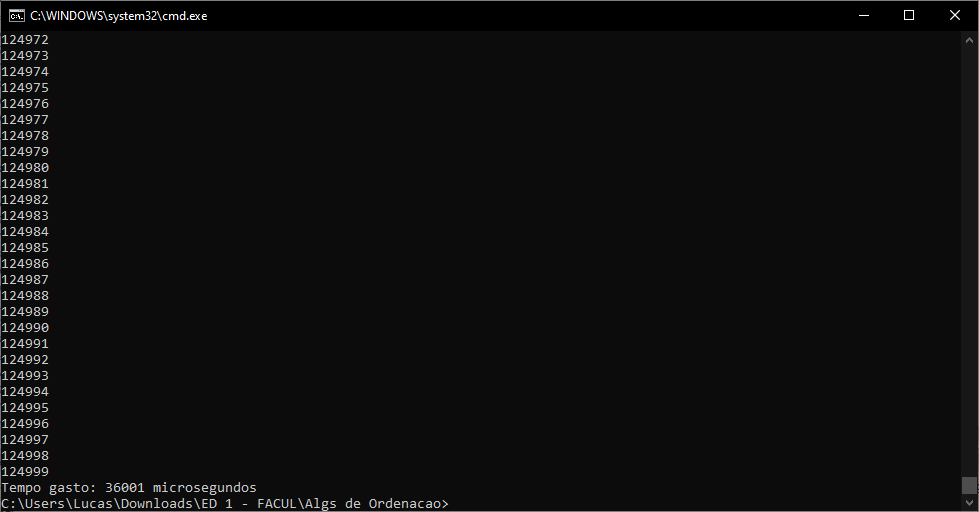
#### **desordenado5k**

1. 
2. 
3. 
4. 
5. 

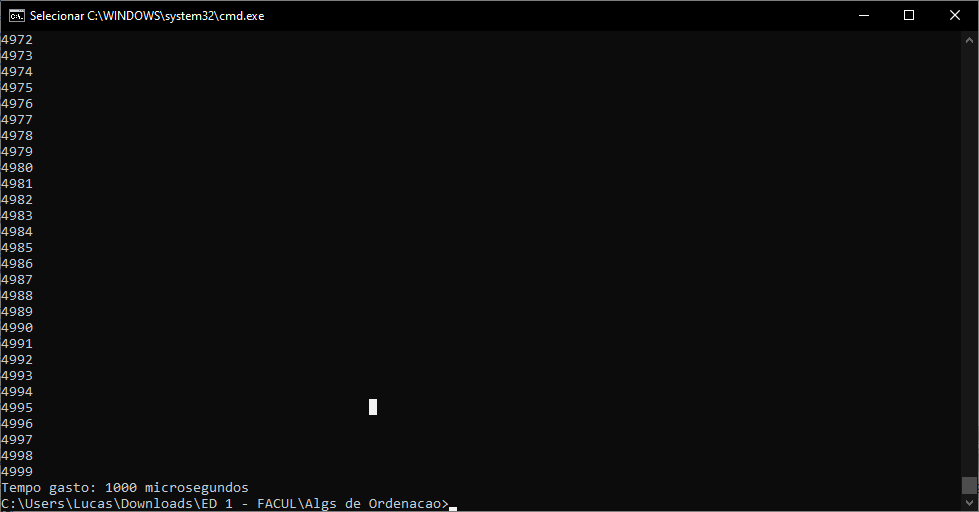
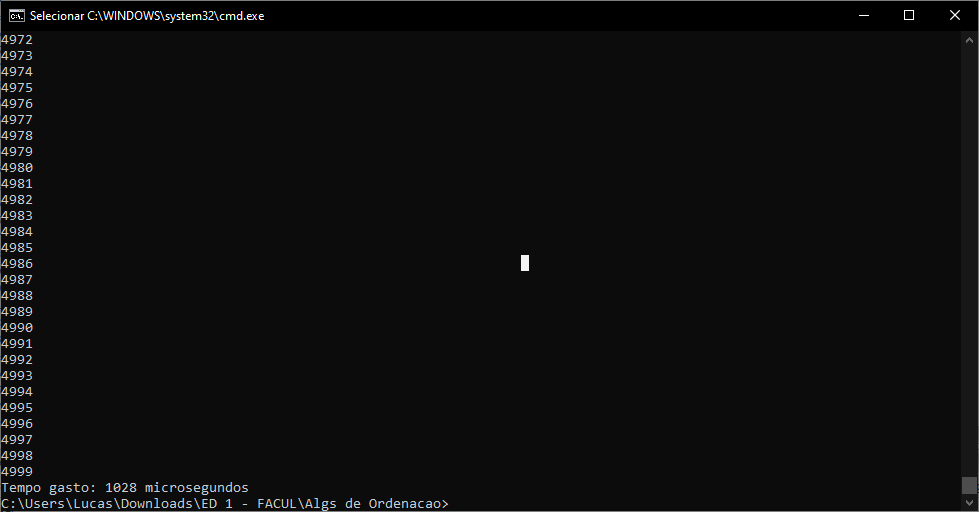
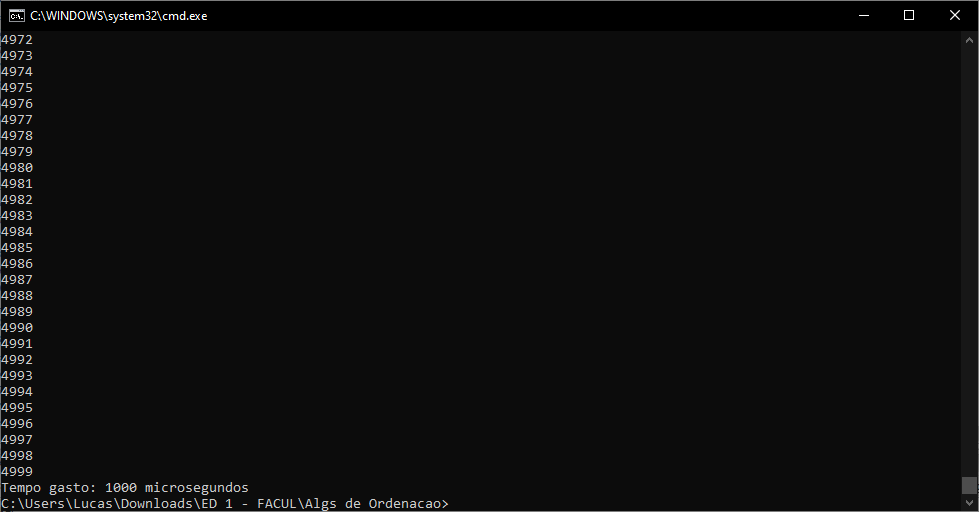
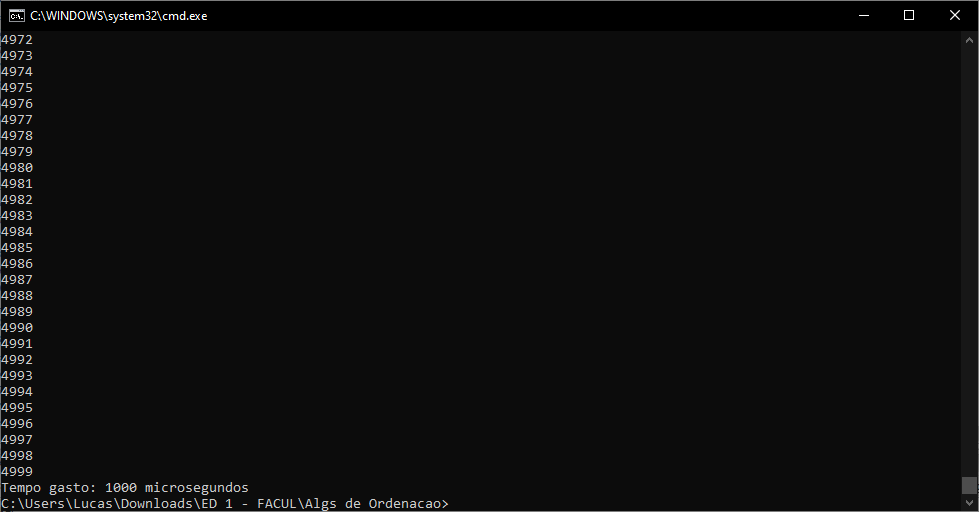
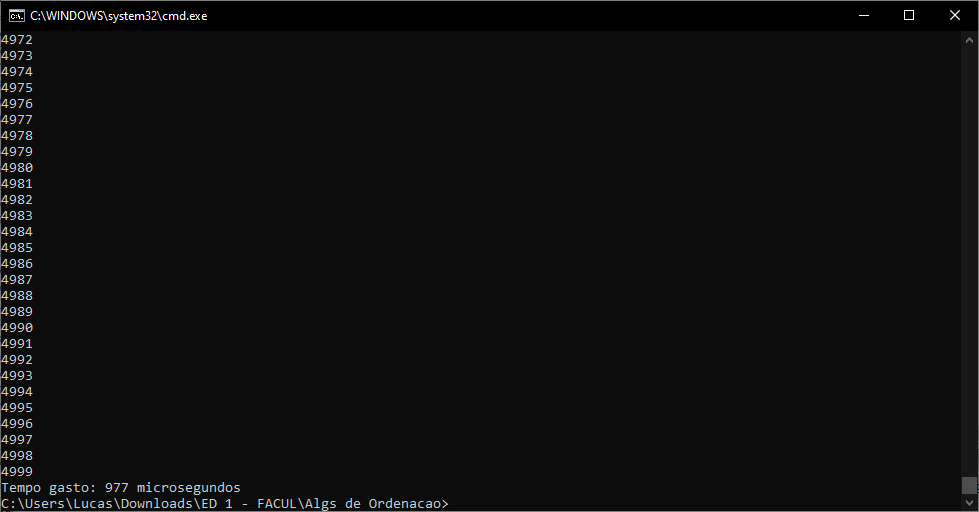
#### **desordenado25k**

1. 
2. 
3. 
4. 
5. 

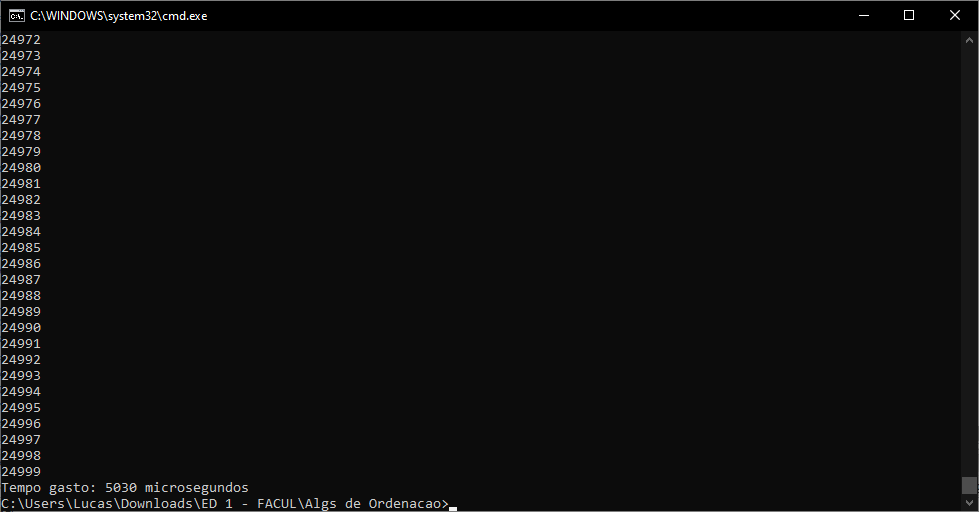
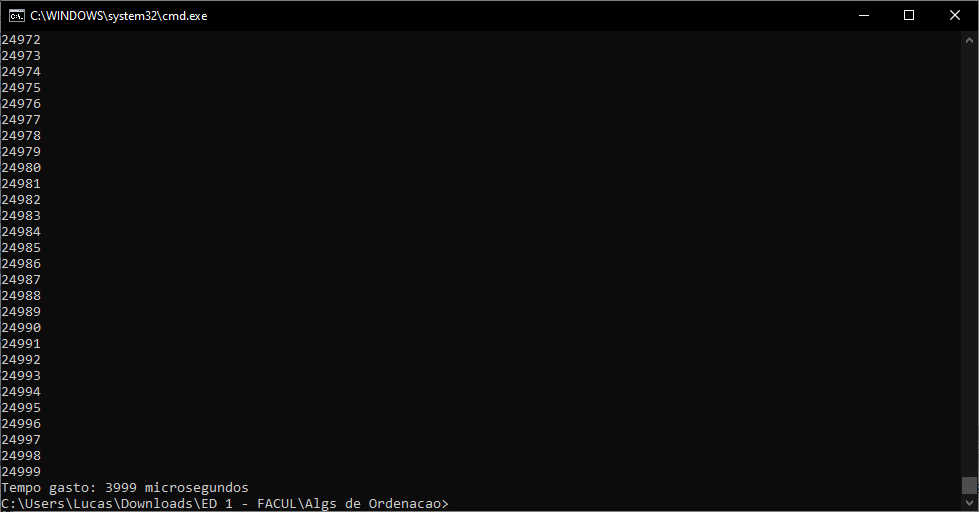
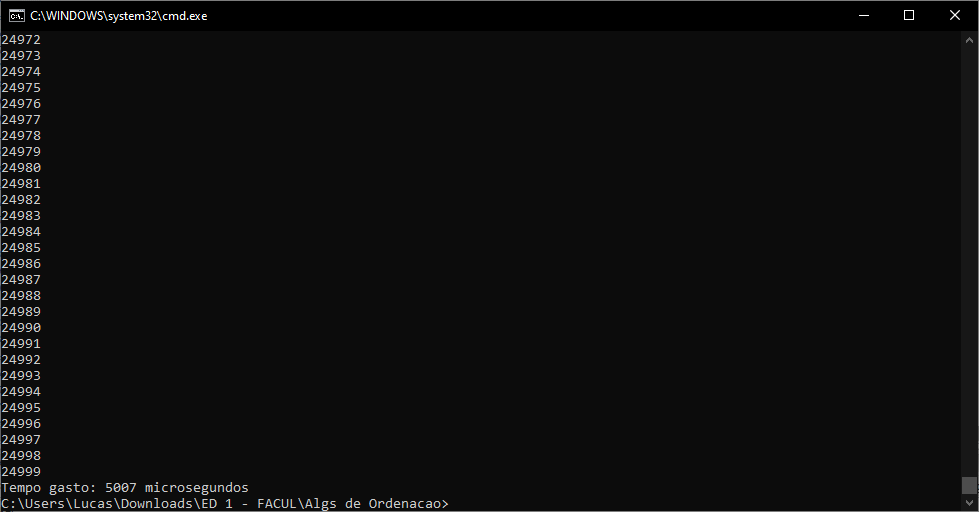
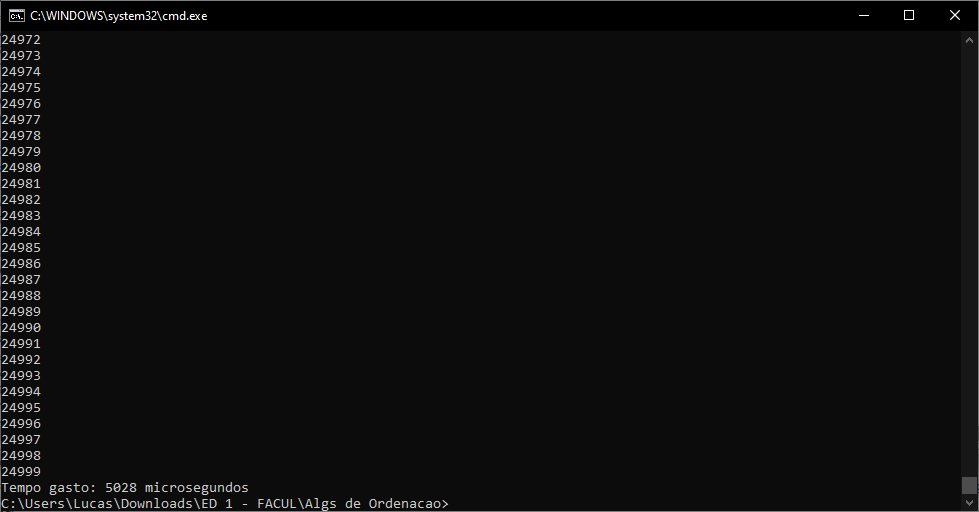
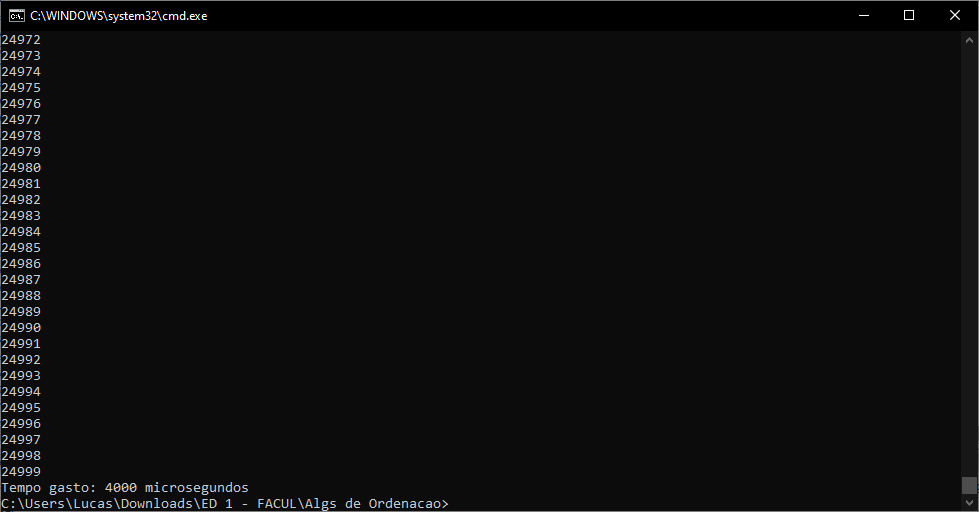
#### **desordenado125k**

1. 
2. 
3. 
4. 
5. 

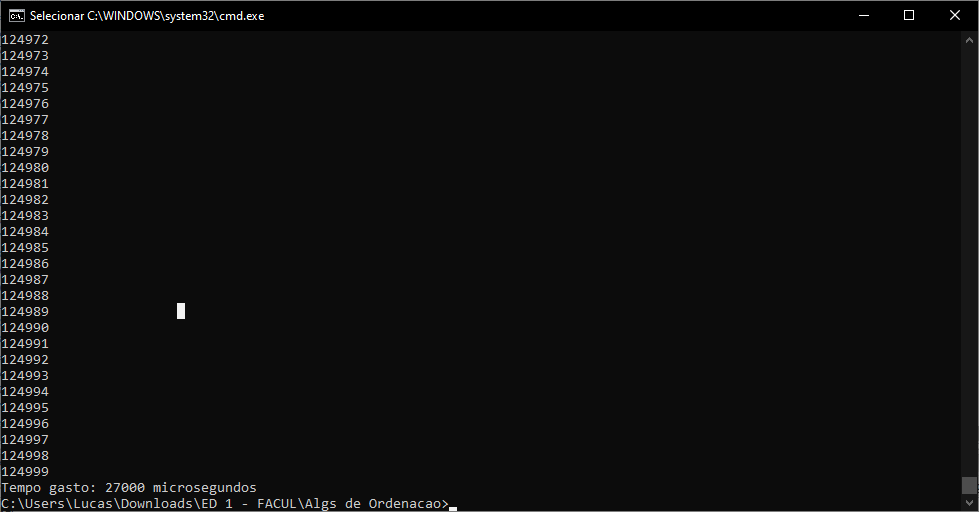
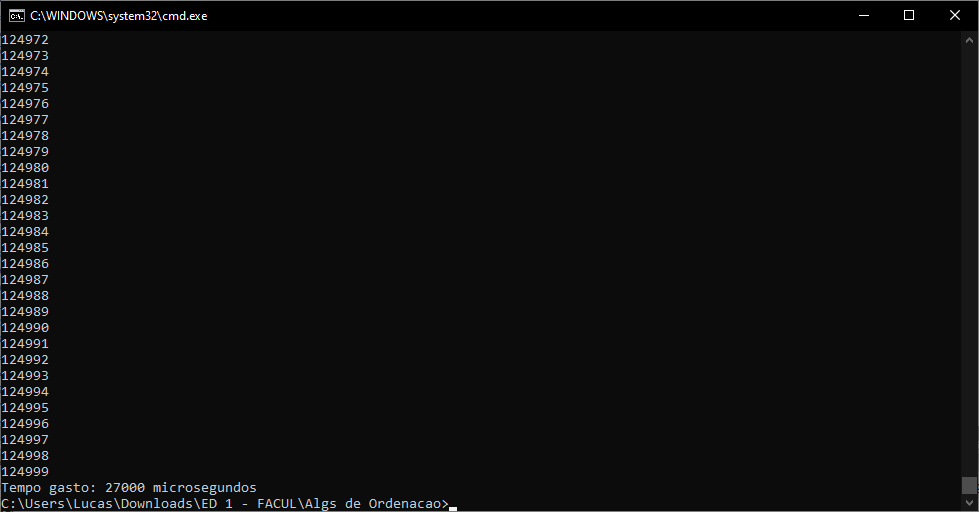
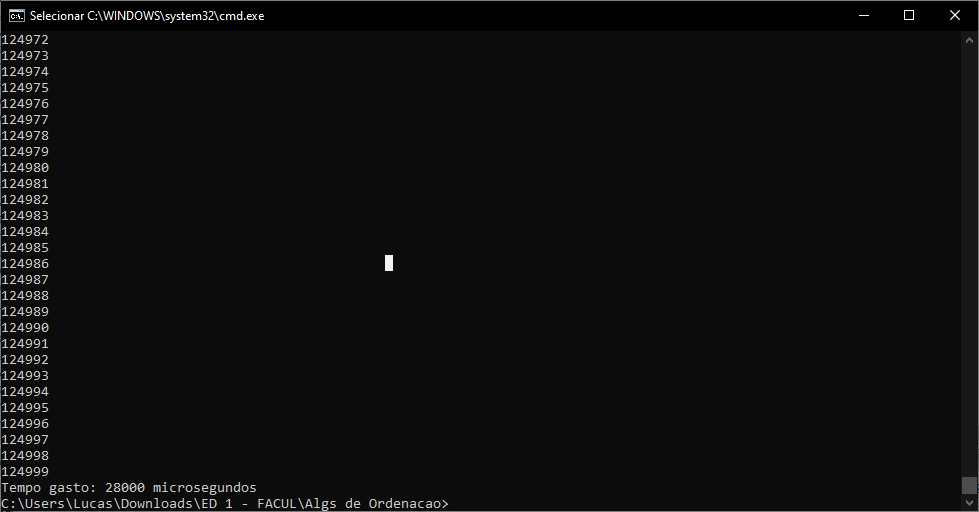
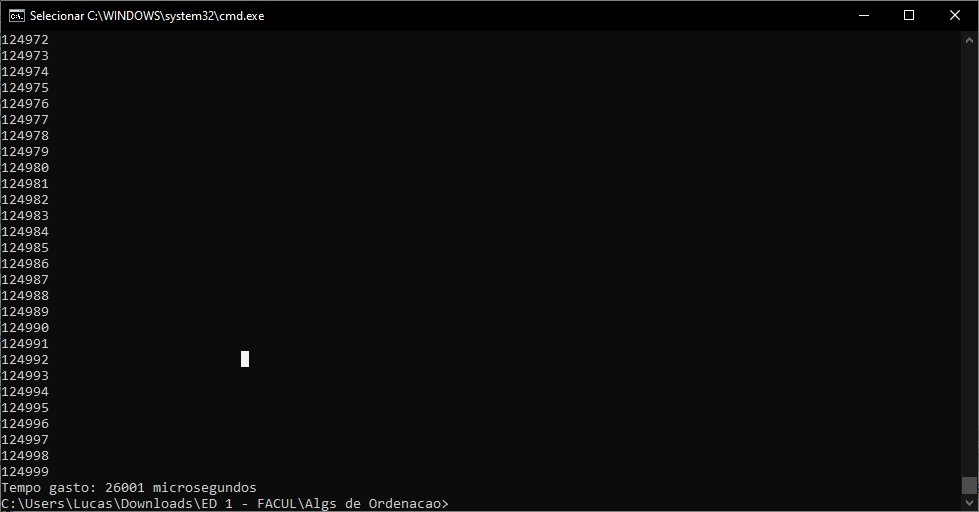
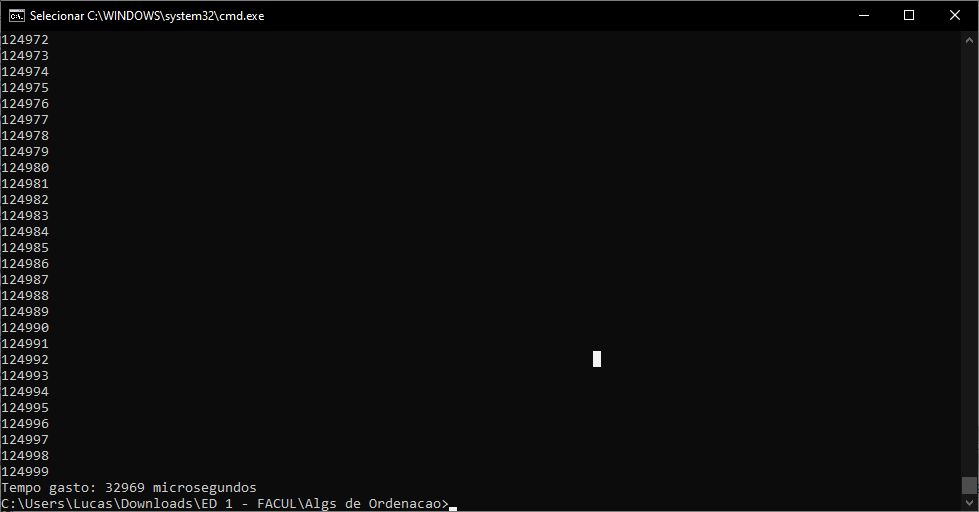
#### **ordenado5k**

1. 
2. 
3. 
4. 
5. 

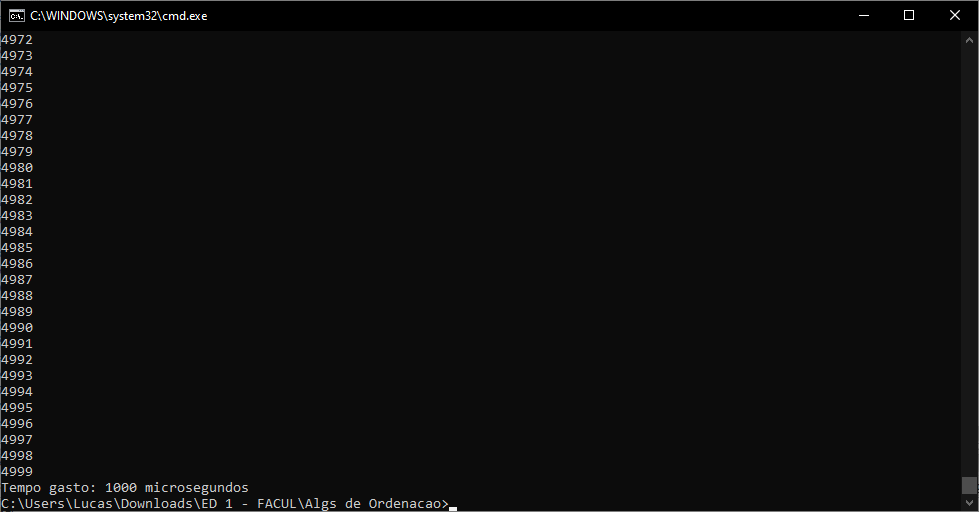
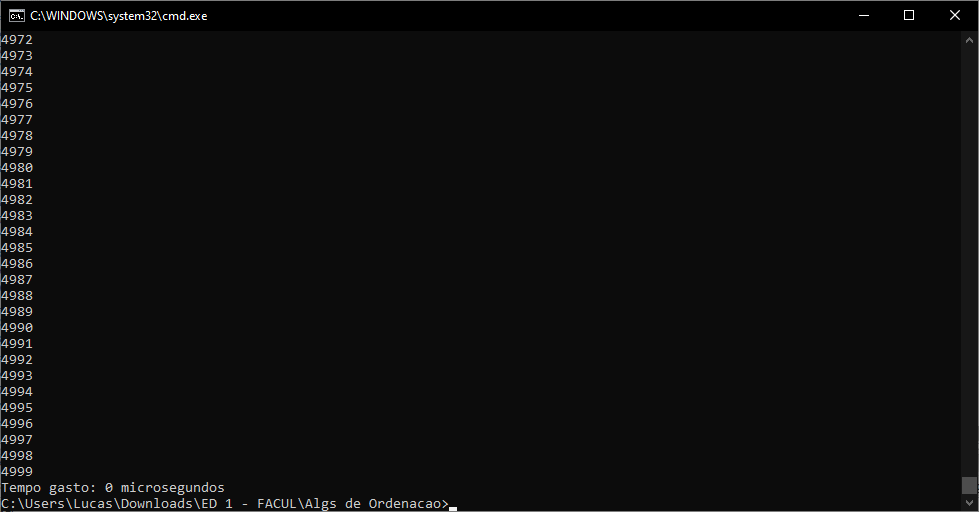
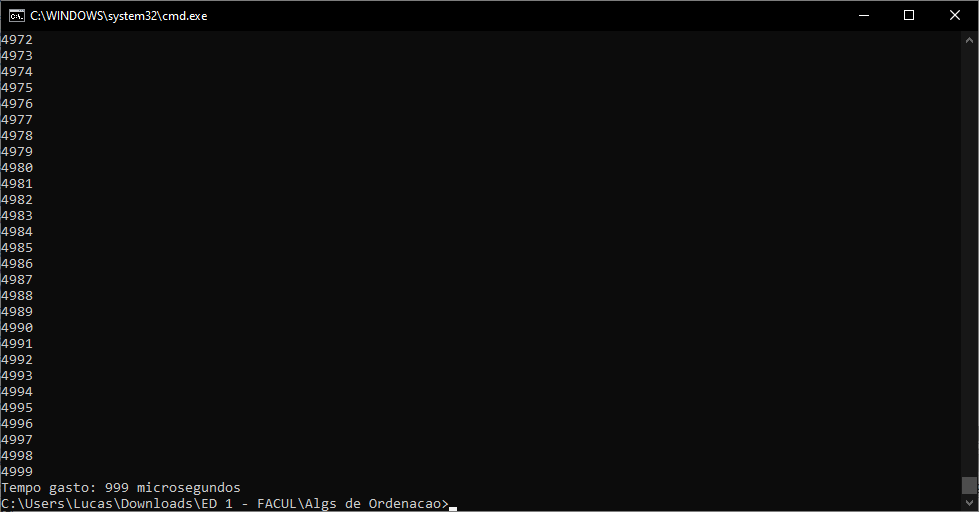
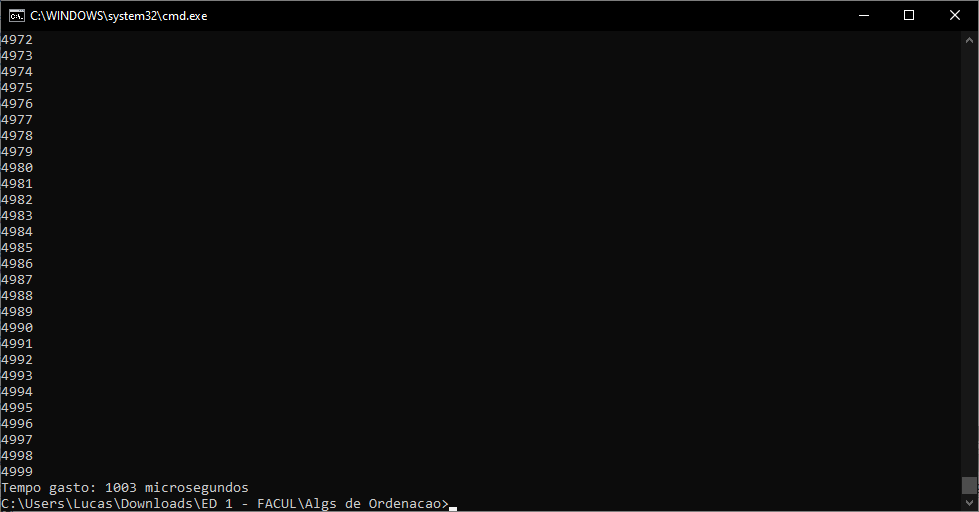
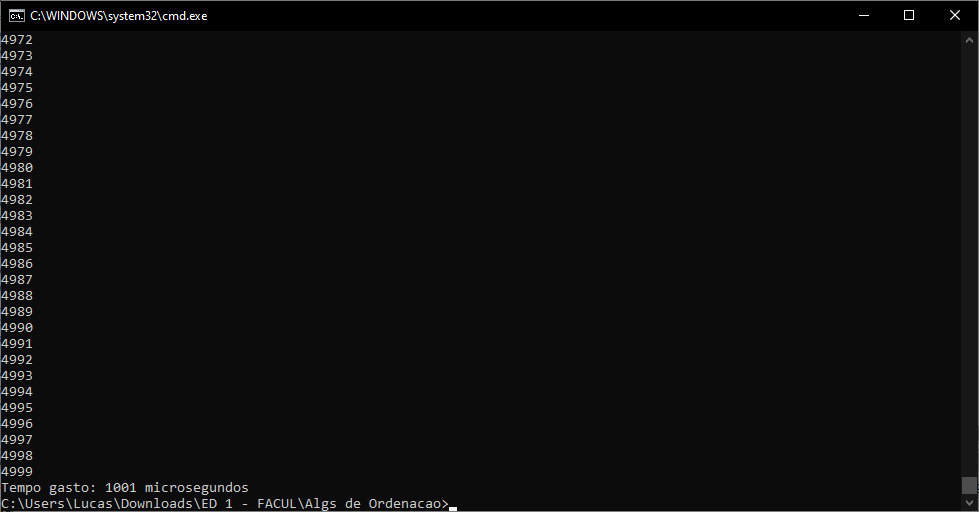
#### **ordenado25k**

1. 
2. 
3. 
4. 
5. 

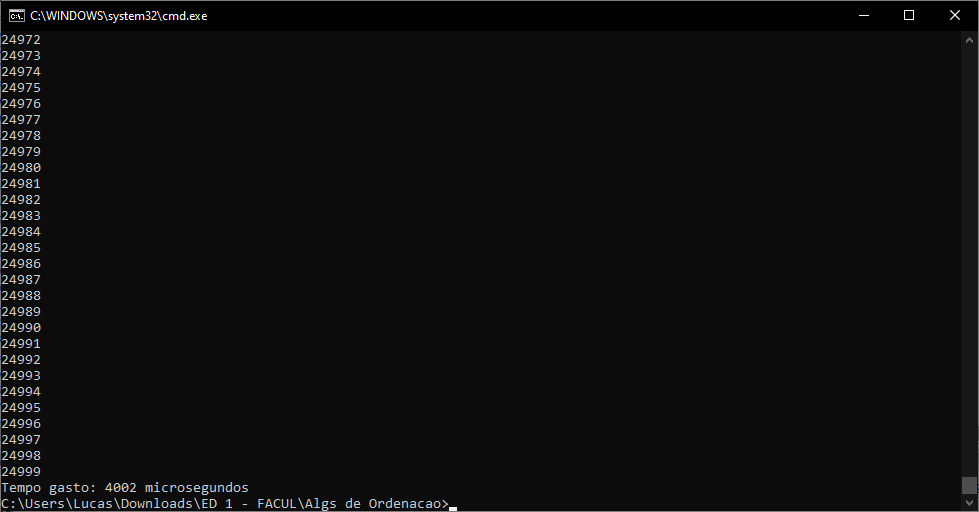
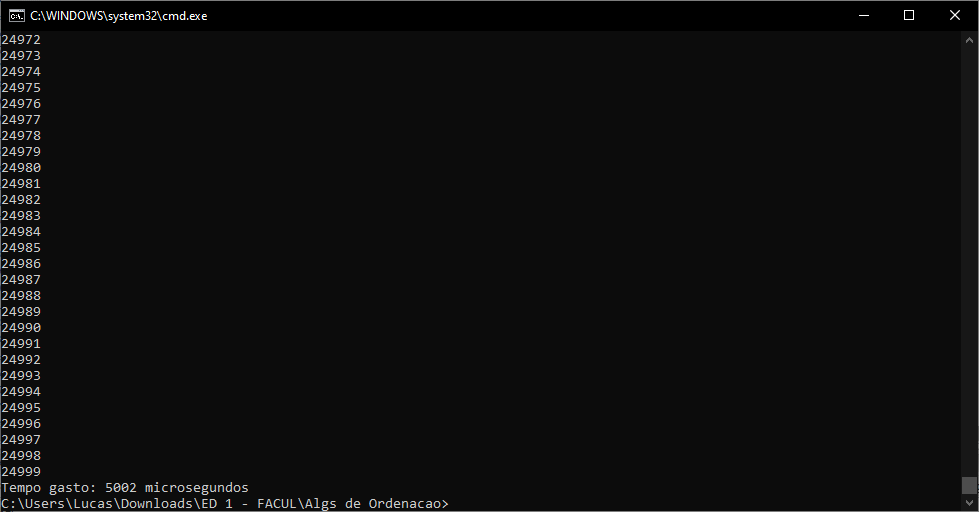
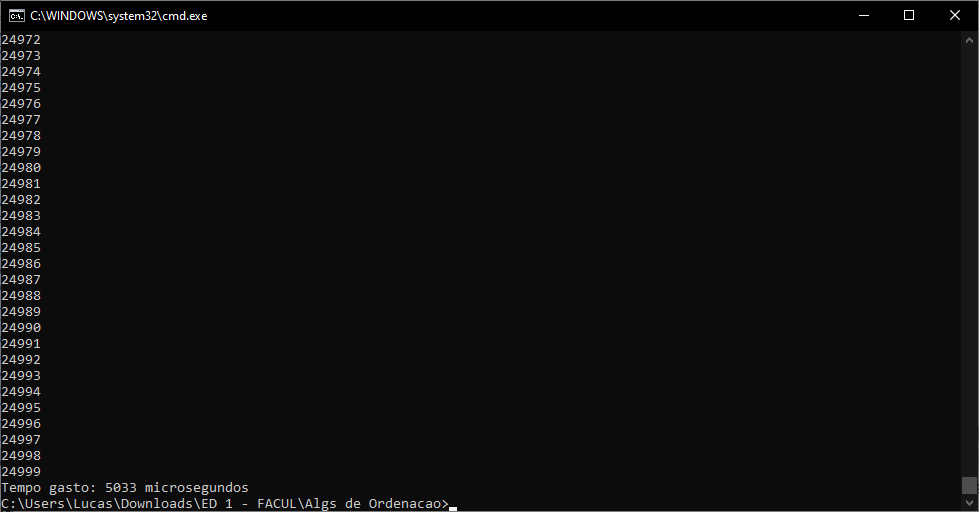
#### **ordenado125k**

1. 
2. 
3. 
4. 
5. 

#### **ordenadodesc5k**

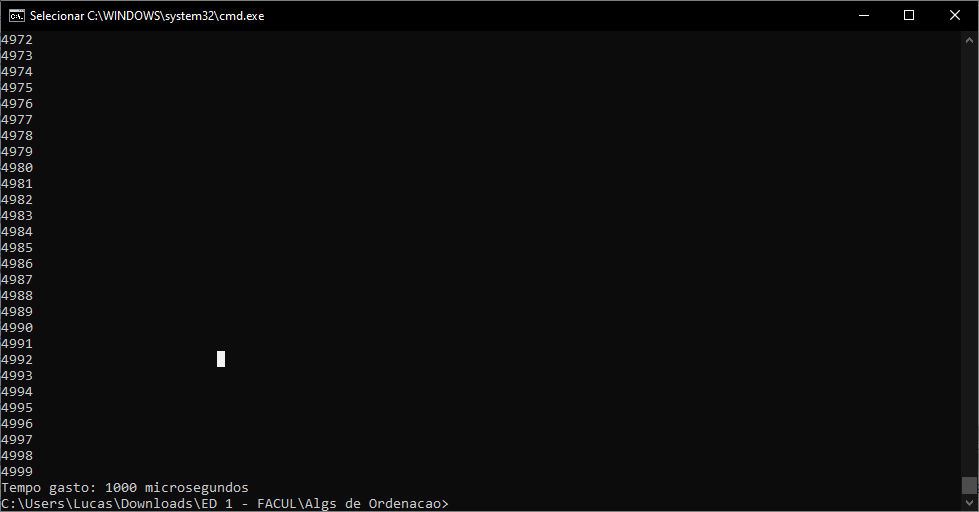
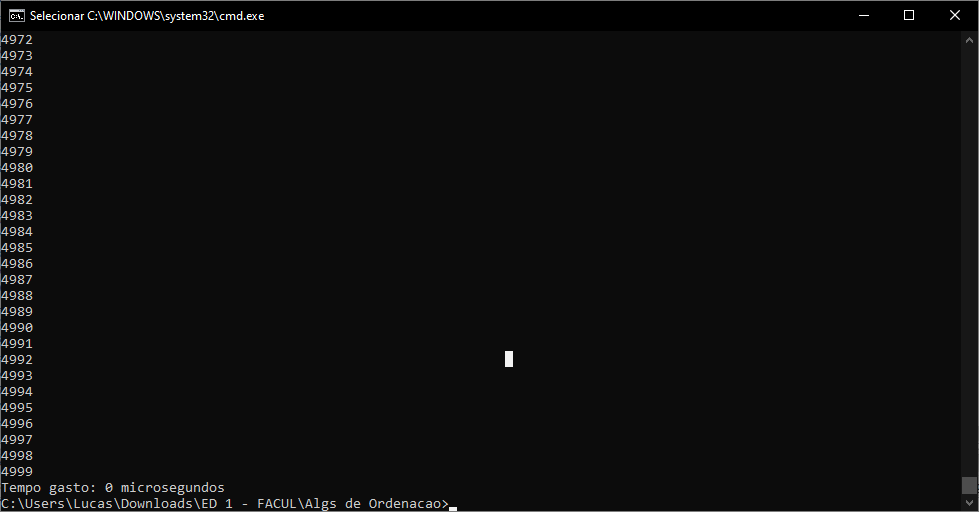
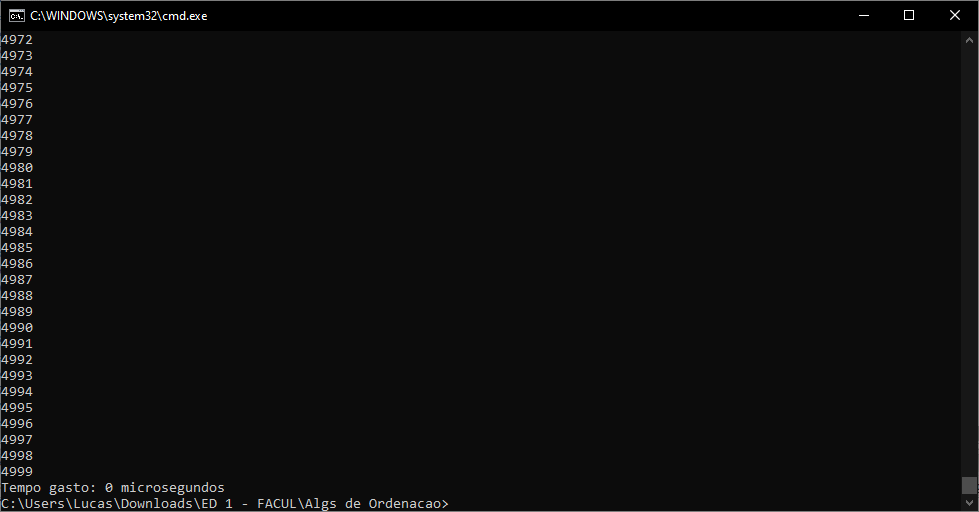
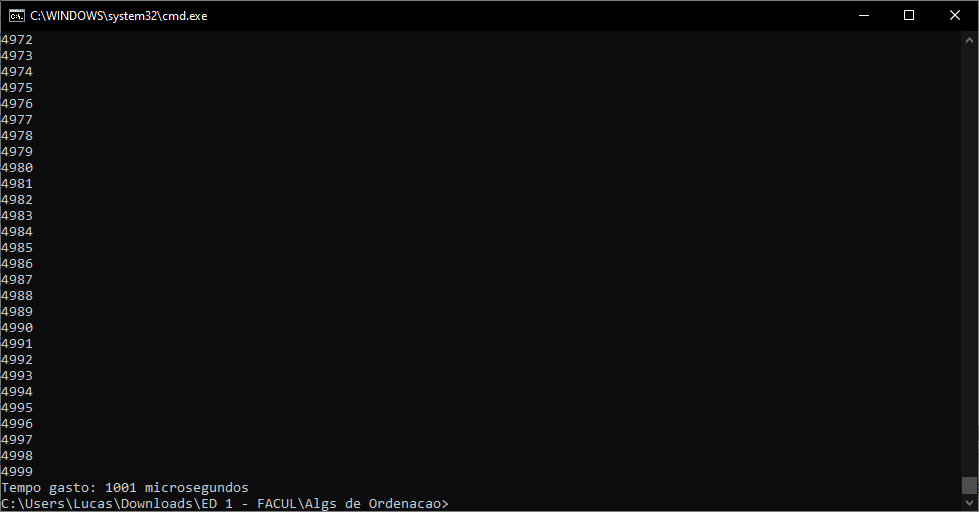
1. 
2. 
3. 
4. 
5. 

#### **ordenadodesc25k**

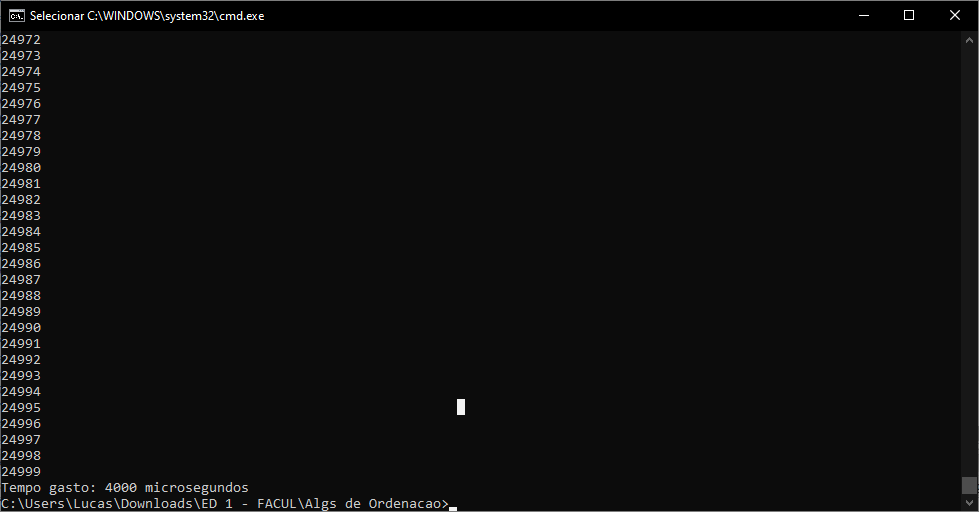
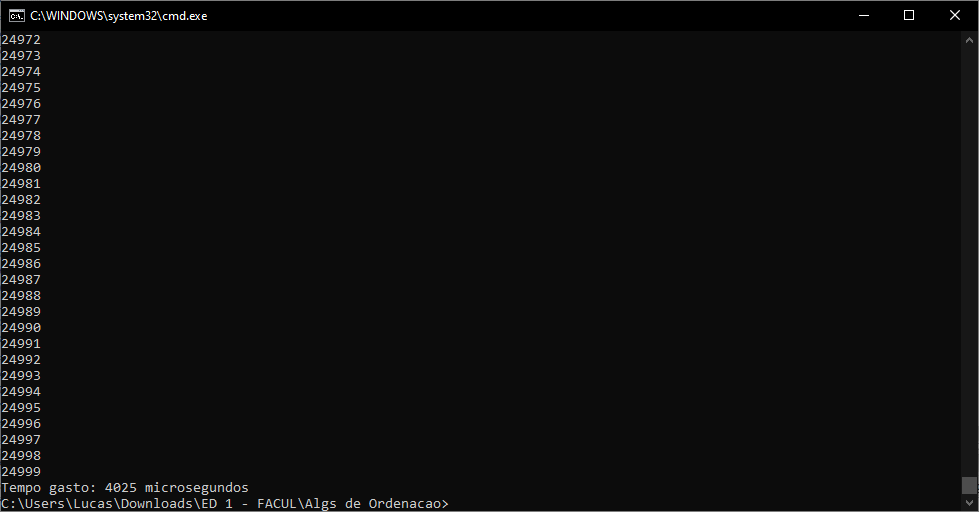
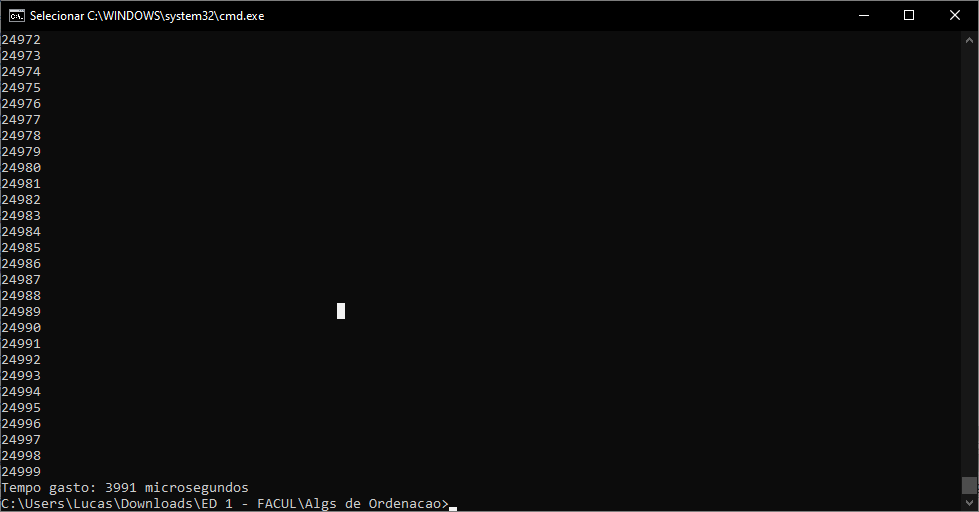
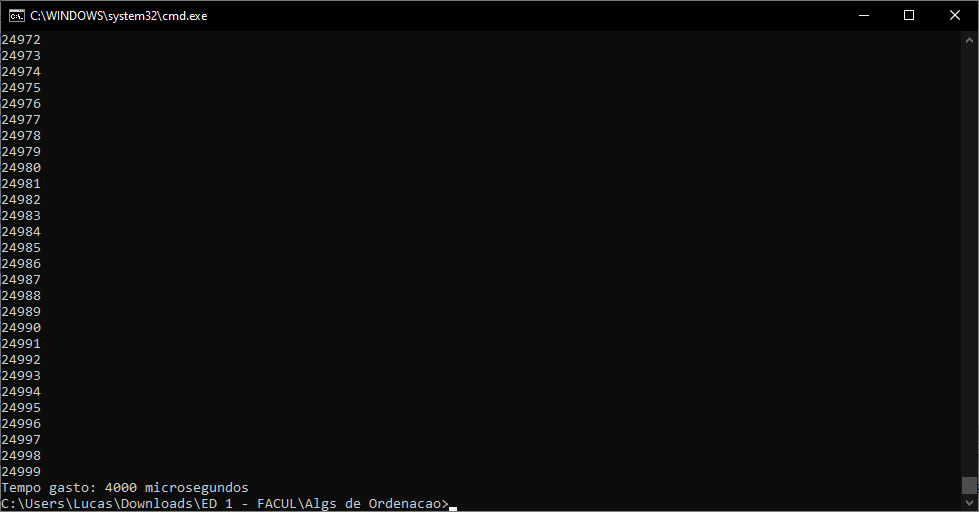
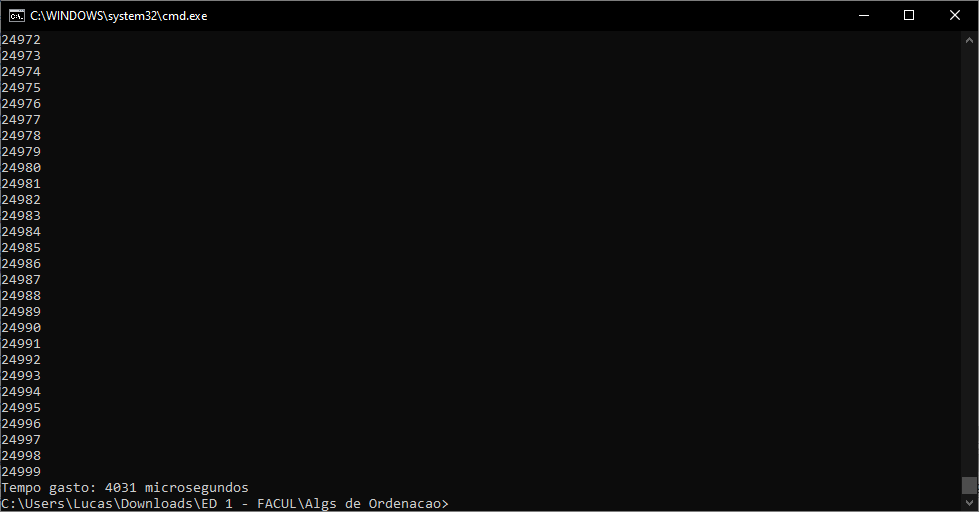
1. 
2. 
3. 
4. 
5. 

### ***QuickSort***

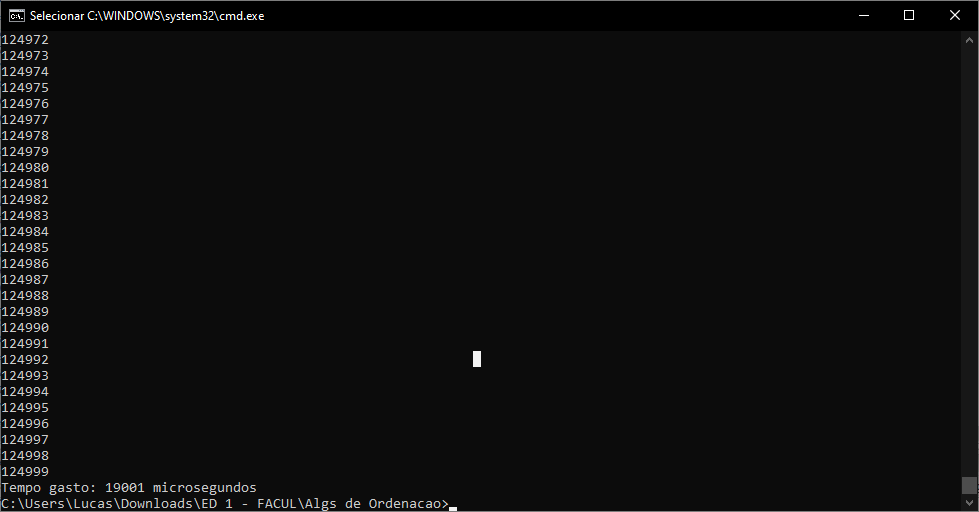
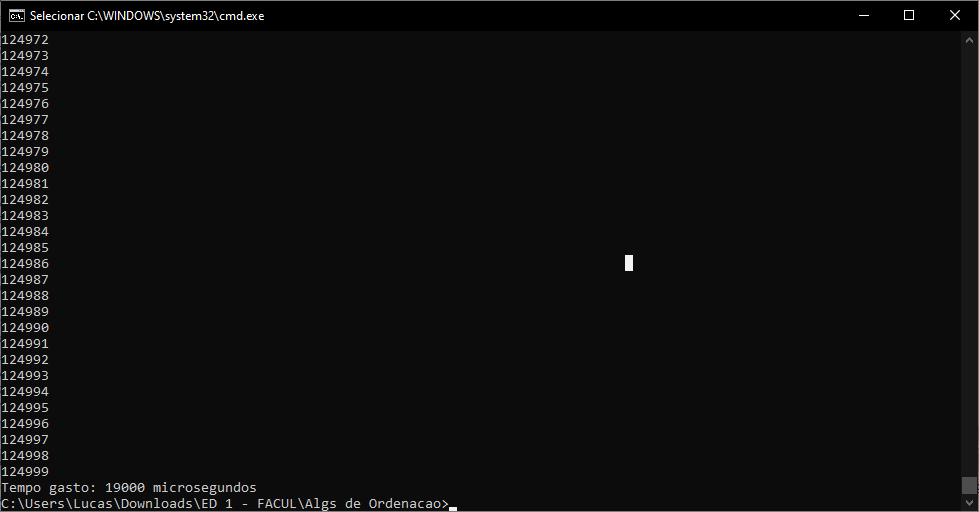
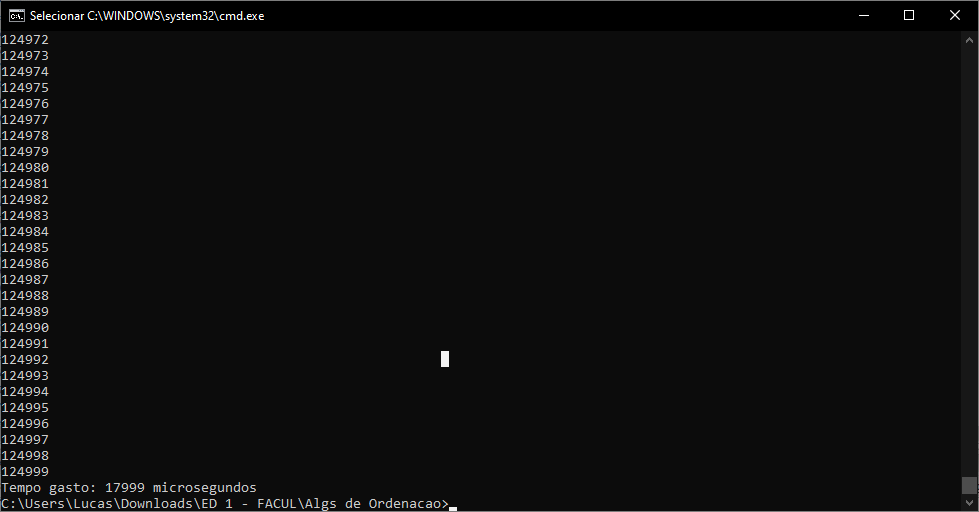
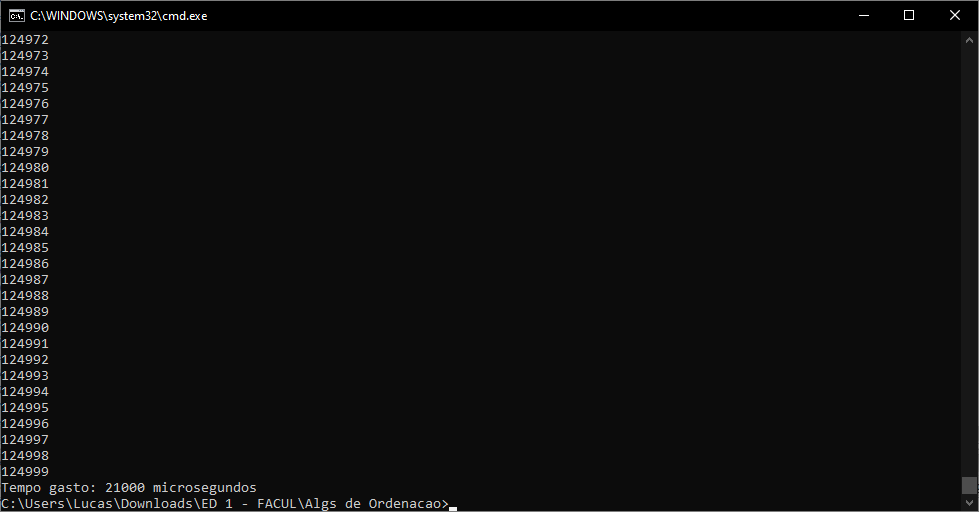
#### ***desordenado5k***

1. 
2. 
3. 
4. 
5. 

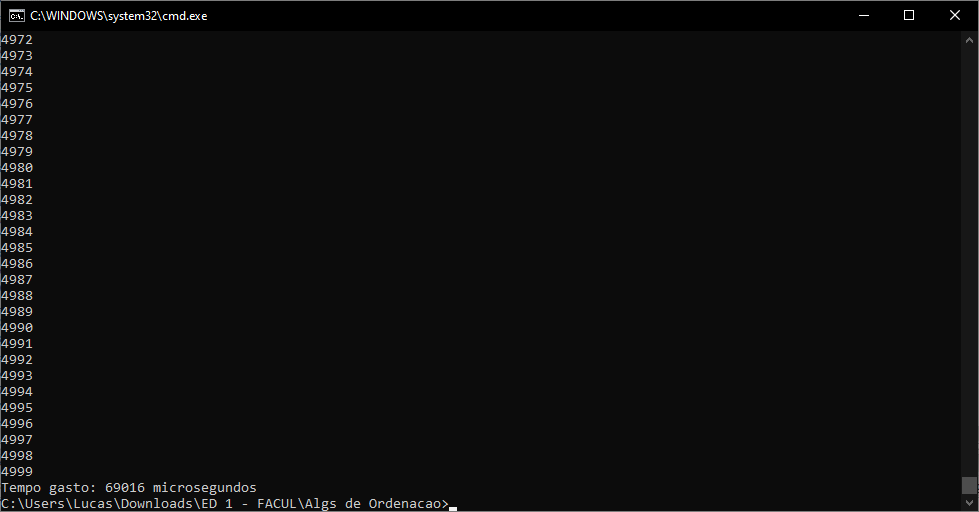
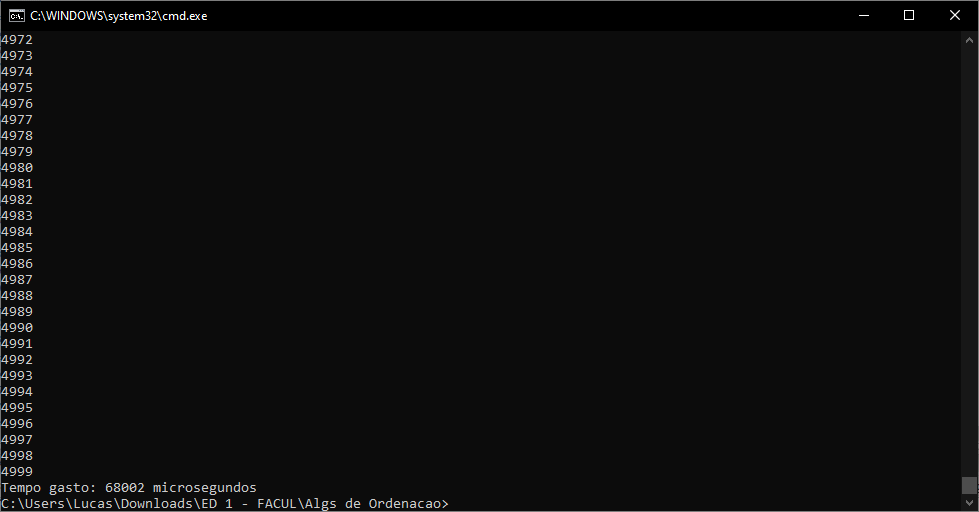
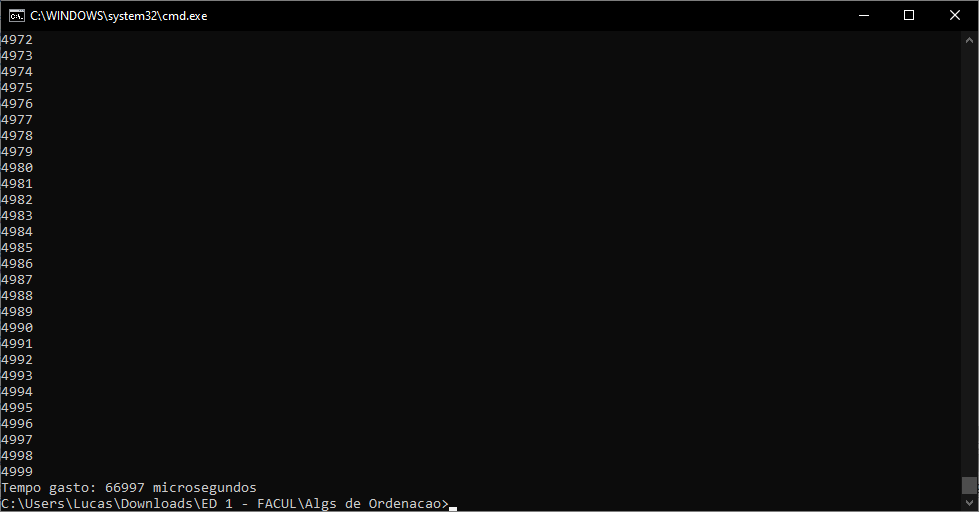
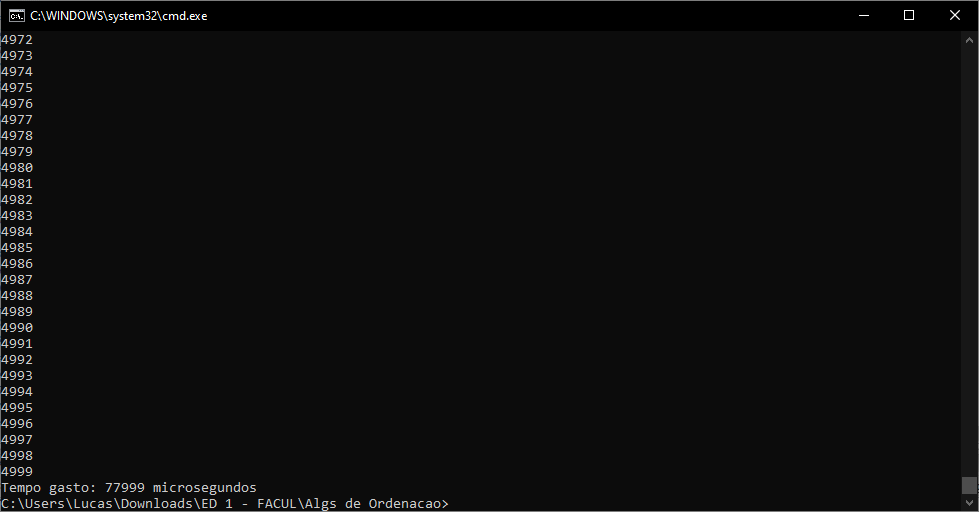
#### ***desordenado25k***

1. 
2. 
3. 
4. 
5. 

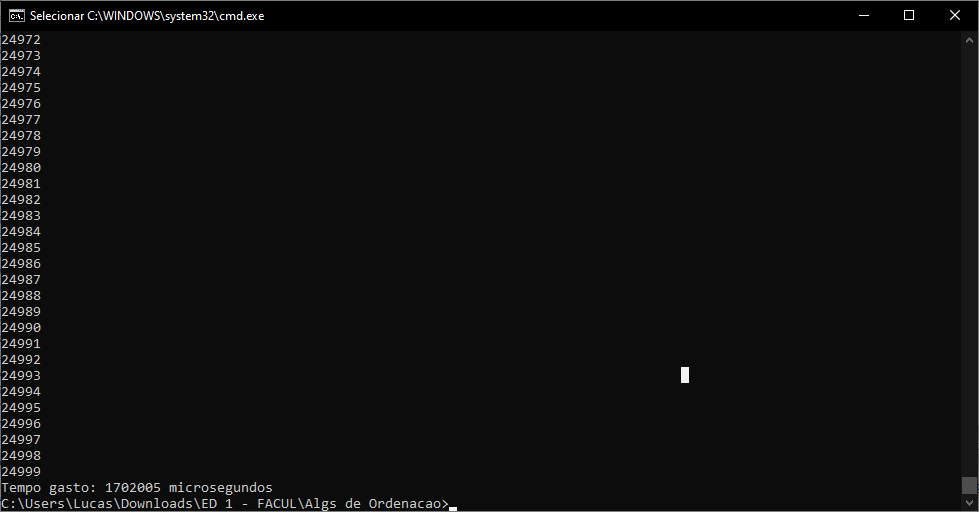
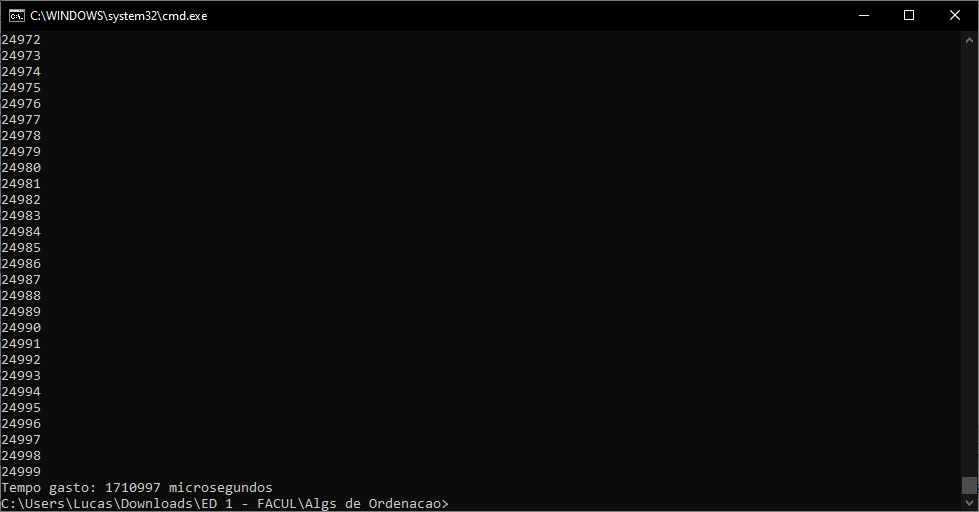
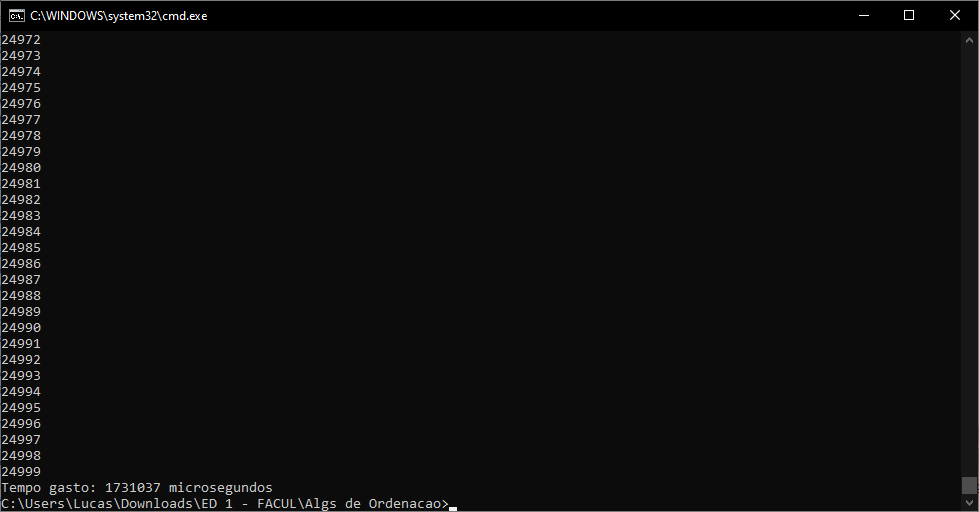
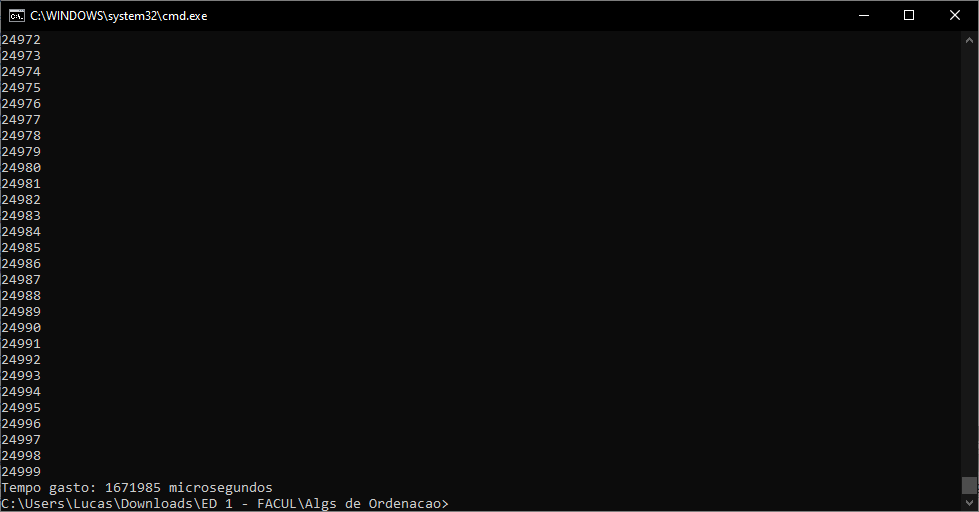
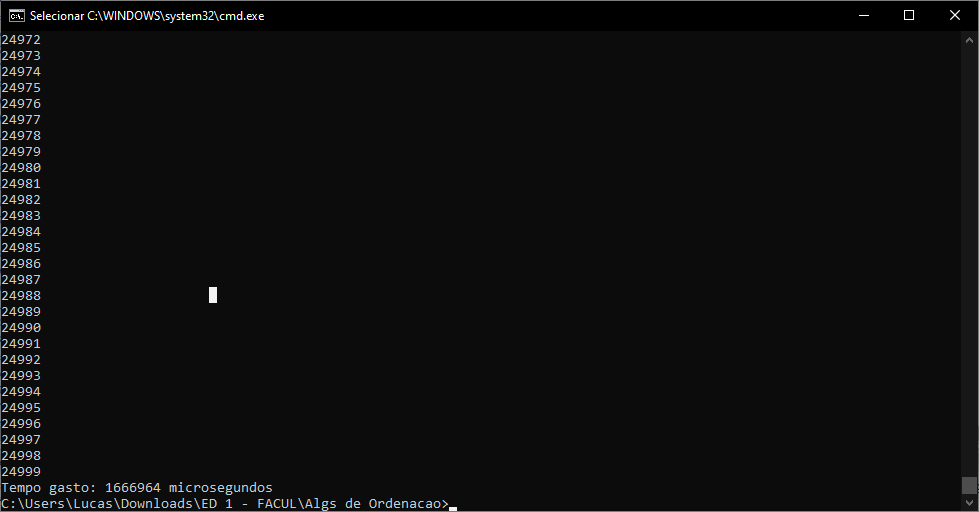
#### ***desordenado125k***

1. 
2. 
3. 
4. 
5. 

#### ***ordenado5k***

1. 
2. 
3. 
4. 
5. 

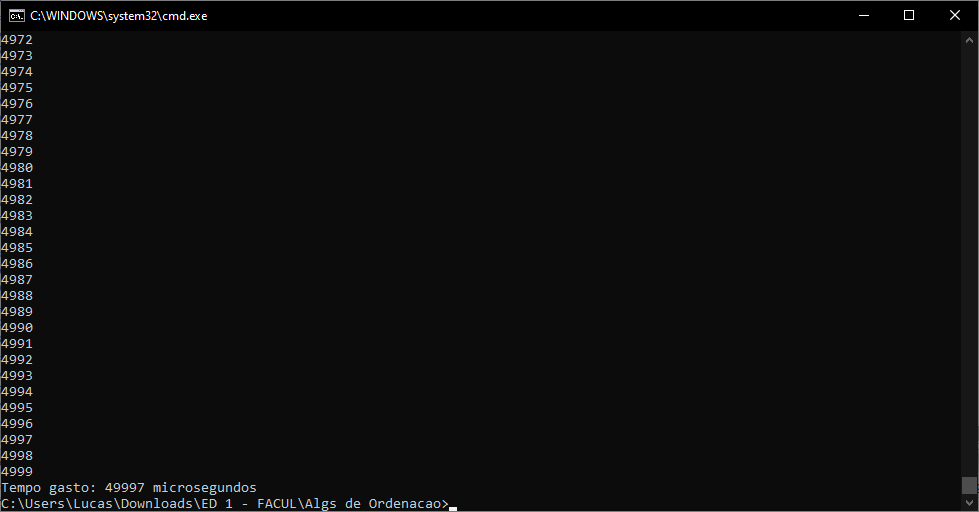
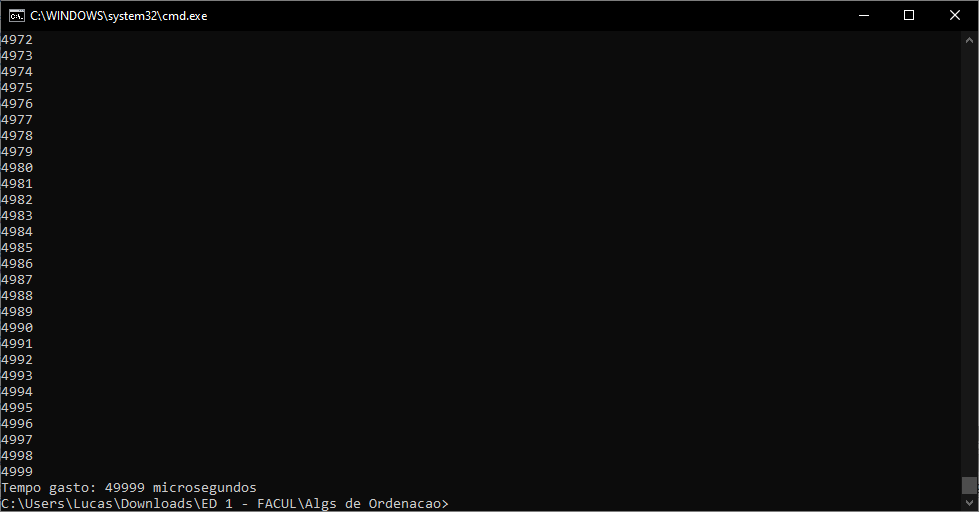
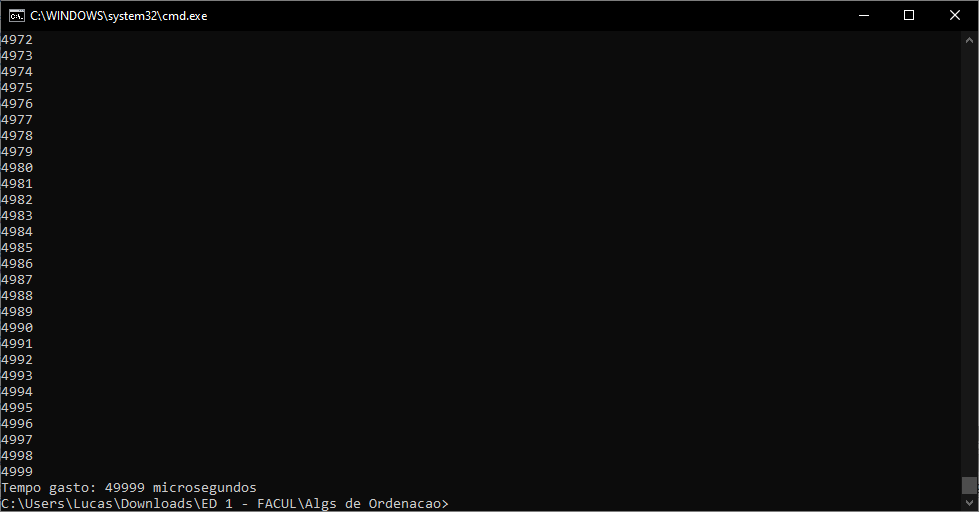
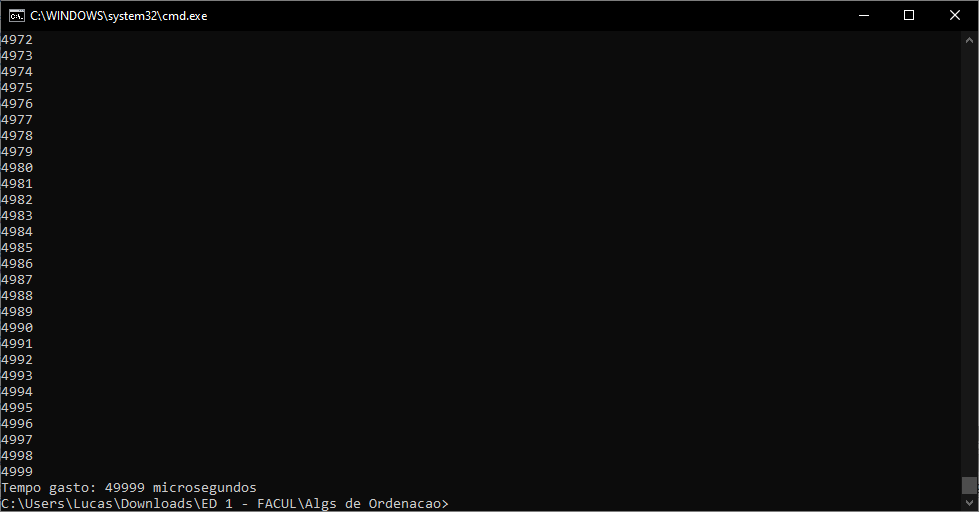
#### ***ordenado25k***

1. 
2. 
3. 
4. 
5. 

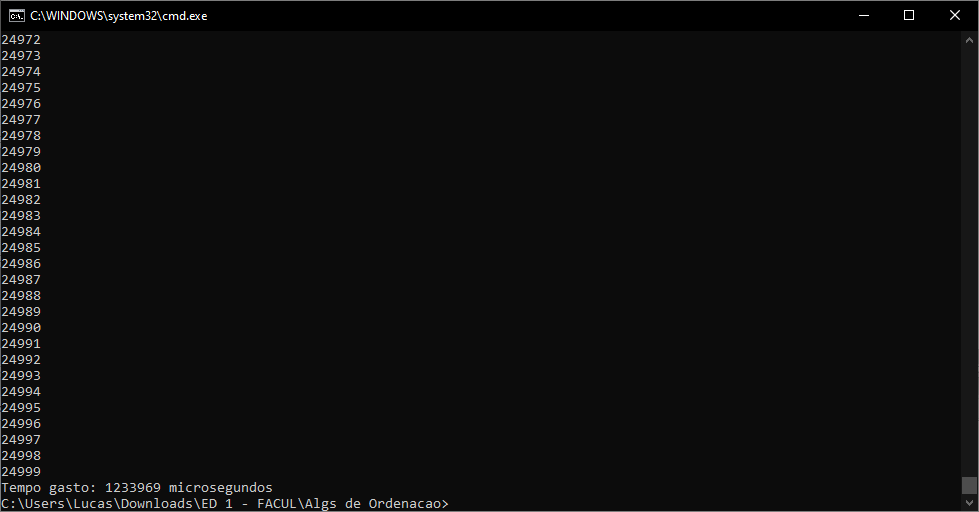
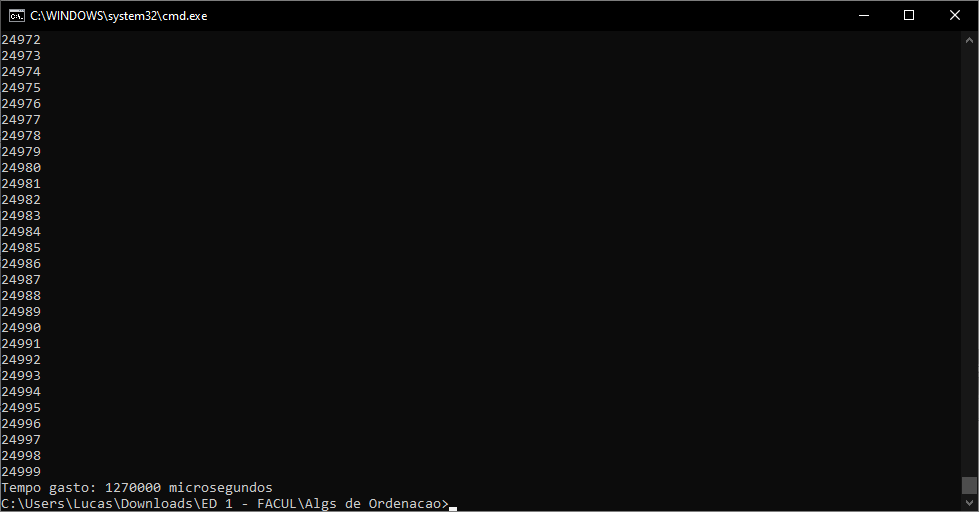
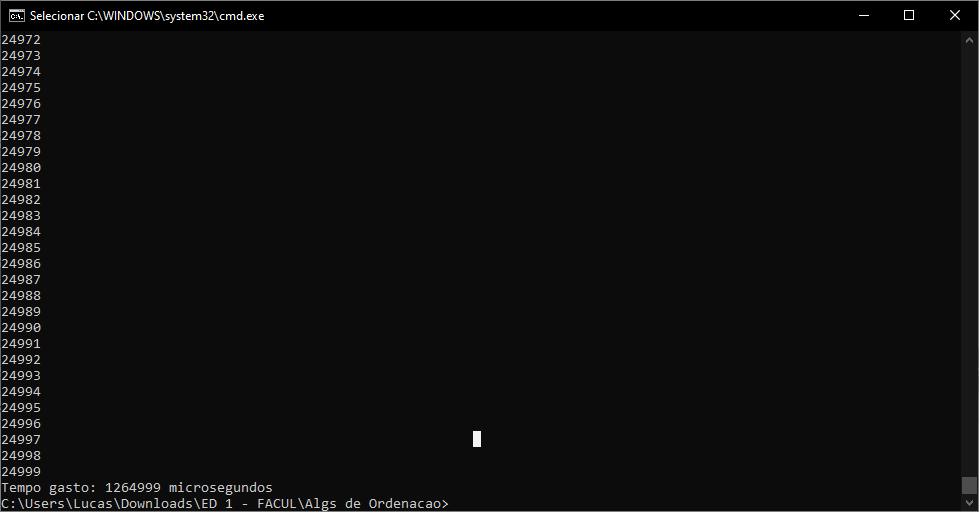
#### ***ordenado125k***

1. d

#### ***ordenadodesc5k***

1. 
2. 
3. 
4. 
5. 

#### ***ordenadodesc25k***

1. 
2. 
3. 
4. 
5. 