

---

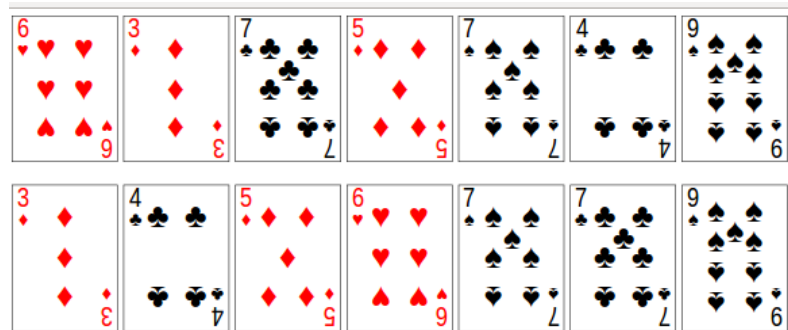
# Ordenação

Prof. Karl Apaza Agüero

# Ordenação

---

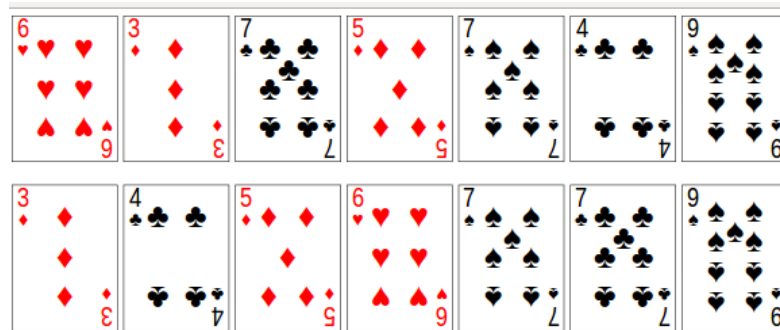
- ▶ Rearranjar elementos em uma ordem específica
  - ▶ ordem numérica crescente
  - ▶ ordem numérica decrescente
  - ▶ ordem alfabética



# Menor número em um vetor

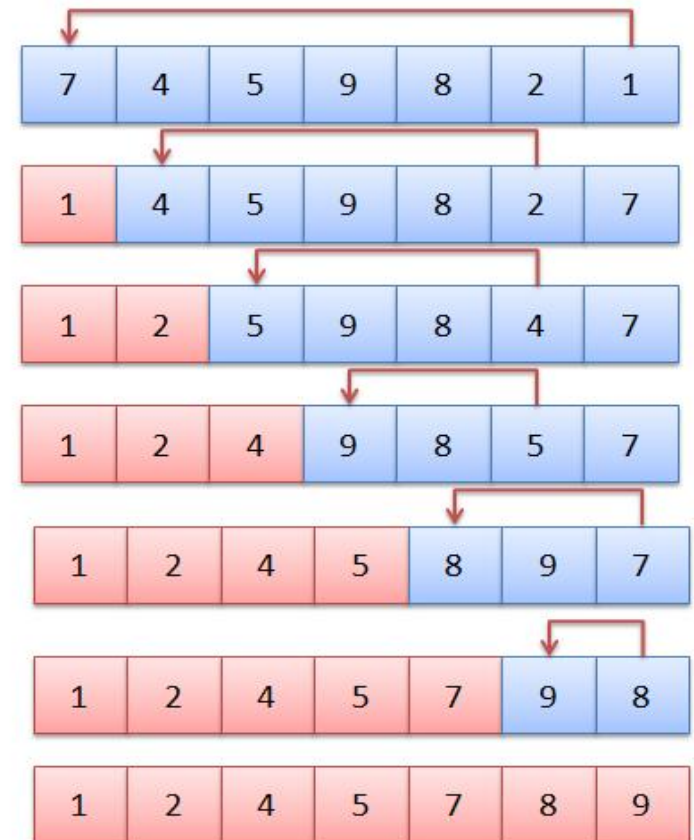
---

```
int vetor[N];  
int menor_i = 0;  
for(int i = 1; i < N; i++){  
    if(vetor[i] < vetor[menor_i]){  
        menor_i = i;  
    }  
}
```



# Algoritmo de Ordenação

- ▶ Selection Sort
  - ▶ Sem vetor auxiliar, usando somente comparação e troca de elementos
  - ▶ Para  $n-1$  trocas (onde  $n$  é o tamanho do vetor)
    - trocar o menor elemento da sequência não ordenada (sequência azul) com o elemento da primeira posição da sequência não ordenada

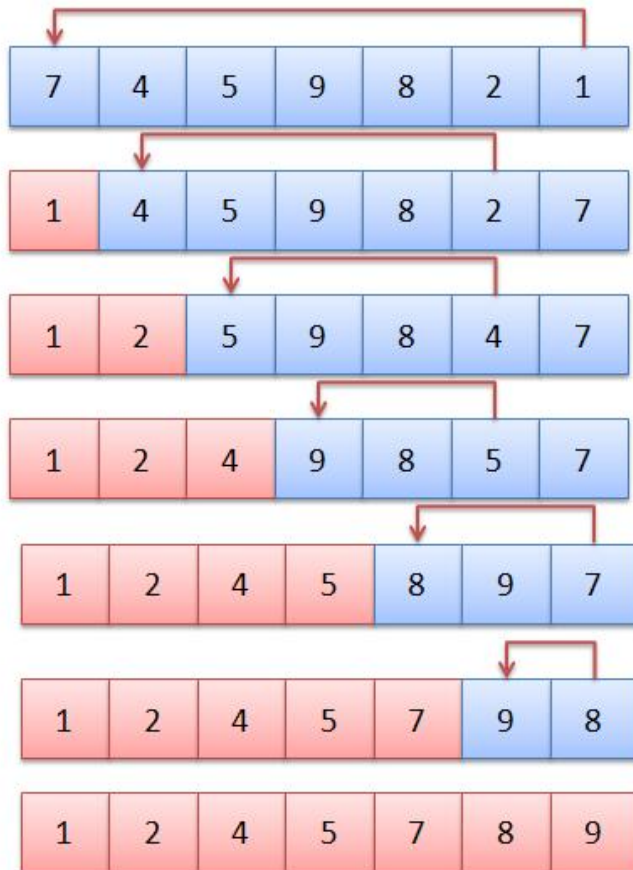


# Exemplo

---

- ▶ Descrição
  - ▶ Imprime uma sequência de números em ordem **crescente**.
- ▶ Entrada
  - ▶ Lê um inteiro  $n$  que indica a quantidade de elementos, e então lê  $n$  elementos desordenados.
- ▶ Saída
  - ▶ Uma linha com os  $n$  elementos ordenados separados por espaço

# Ordenação



```
#include <stdio.h>
```

```
int main() {
```

```
    int i, j, N, menor_i, aux;
```

```
    scanf("%d", &N);
```

```
    int vetor[N];
```

```
    // Leitura dos N elementos
```

```
    for(i=0; i < N; i++)
```

```
        scanf("%d", &vetor[ i ]);
```

```
    for(j = 0; j < N-1; j++) { // Ordenação
```

```
        menor_i = j;
```

```
        for(i = j+1; i < N; i++){
```

```
            if(vetor[ i ] < vetor[menor_i]){
```

```
                menor_i = i;
```

```
            }
```

```
        }
```

```
        aux = vetor[ j ];
```

```
        vetor[ j ] = vetor[menor_i];
```

```
        vetor[menor_i] = aux;
```

```
    }
```

```
    for (i=0; i < N-1; i++) // Impressão
```

```
        printf("%d ", vetor[ i ]);
```

```
    printf("%d\n", vetor[ i ]);
```

```
    return 0;
```

```
}
```

## Problema

Bebe-bebe é um jogo muito popular de fliperama. E, como a maioria dos jogos de fliperama, ele deve mostrar as maiores pontuações. Para esse fim, a companhia Otori te contratou.

Escreva um programa que, dada a lista de todas as pontuações dos jogos de Bebe-bebe, mostra os melhores placares em ordem decrescente.

## Entrada

A entrada é composta de um único caso de teste. A primeira linha consiste de dois inteiros N e M, dizendo quantas partidas foram jogadas de Bebe-bebe e quantas linhas cabem no mostrador de melhores rankings. As N linhas seguintes contêm cada uma um inteiro indicando a pontuação obtida em cada jogo.

## Saída

Seu programa deve imprimir M linhas, contendo as M maiores pontuações em ordem decrescente.

## Entrada

7 4

100

200

200

150

30

524

942

## Saída

942

524

200

200

```
#include <stdio.h>
int main() {
    int i, j, N, M, maior_i, aux;
    scanf("%d %d", &N, &M);
    int vetor[N];

    for(i=0; i < N; i++) scanf("%d", &vetor[ i ]);

    for(j = 0; j < M; j++) {
        maior_i = j;
        for(i = j+1; i < N; i++){
            if(vetor[ i ] > vetor[maior_i]){
                maior_i = i;
            }
        }
        aux = vetor[ j ];
        vetor[ j ] = vetor[maior_i];
        vetor[maior_i] = aux;
    }
    for (i=0; i < M; i++) printf("%d\n", vetor[ i ]);
    return 0;
}
```



# Problema

Escrever um programa para determinar, a partir de uma listagem de aeroportos e vôos, qual aeroporto possui maior probabilidade de congestionamento. Como medida da probabilidade de congestionamento será utilizado neste estudo o número total de vôos que chegam ou que partem de cada aeroporto.

**Entrada:** A primeira linha contém dois números inteiros A e V, que indicam respectivamente o número de aeroportos e o número de vôos. Os aeroportos são identificados por inteiros de 1 a A. As V linhas seguintes contêm cada uma a informação de um vôo, representada por um par de números inteiros positivos X e Y, indicando que há um vôo do aeroporto X para o aeroporto Y.

**Saída:** Seu programa deve imprimir os identificadores dos aeroportos em ordem decrescente de vôos. Garante-se que todos os vôos acumulados em cada aeroporto são valores diferentes.

## Exemplo

Entrada:

```
4 5
1 3
2 1
3 2
2 3
3 4
```

Saída:

```
3 2 1 4
```

```
#include <iostream>
using namespace std;
int main(){
```

---

```
    int a,v; cin>>a>>v;
    int aeroportos[a], id_aeroportos[a];
    for(int i=0; i<a; i++){
        aeroportos[i]=0;
        id_aeroportos[i]=i;
    }
```

```
    int id;
    for(int i=0; i<2*v; i++){
        cin>>id;
        aeroportos[id-1]++;
    }
```

```
    for(int j=0; j<a-1; j++){
        int maior_i=j;
        for(int i=j+1; i<a; i++)
            if(aeroportos[i]>aeroportos[maior_i])
                maior_i=i;
```

```
        int aux = aeroportos[j];
        aeroportos[j]=aeroportos[maior_i];
        aeroportos[maior_i]=aux;
```

```
        aux = id_aeroportos[j];
        id_aeroportos[j]=id_aeroportos[maior_i];
        id_aeroportos[maior_i]=aux;
```

```
    }
```

```
    for(int j=0; j<a; j++)
        cout<<id_aeroportos[j]+1<<" ";
```

```
    return 0;
```

```
}
```