Lista de Exercícios

Bubble Sort e Insertion Sort

1. Execute o algoritmo buble Sort com os seguintes valores: 3 x 10^6 , 5x10^6, 7x10^6, 9x10^6, 11x10^6, 13,x10^6, 15x10^6, 17x10^6, 19x10^6, 21x10^6. Gere os gráficos de tempo para vetor ordenador crescente, vetor ordenado decrescente e vetor com números aleatórios.
2. Execute o algoritmo Insertion Sort com os seguintes valores: 3 x 10^6 , 5x10^6, 7x10^6, 9x10^6, 11x10^6, 13,x10^6, 15x10^6, 17x10^6, 19x10^6, 21x10^6. Gere os gráficos de tempo para vetor ordenador crescente, vetor ordenado decrescente e vetor com números aleatórios.
3. Implemente a **struct filme** com os campos: *id_filme*, *nome_filme*. Teste a ordenação considerando um vetor de **struct filme** ordenado, vetor de **struct filme** ordenado decrescente e um vetor de **struct filme** com valores aleatórios. Qual algoritmo executou mais rápido ? Buble Sort ou Insertion Sort ?
4. Considerando o vetor **struct filme** aleatório, faça a busca *sequencial* e *binária* por um *id_filme* e analise o tempo de processamento.

Código: gera 10 números aleatórios com intervalo [0-NUM-1].

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

#define NUM 100

int main() {
   srand((unsigned int)time(NULL));
   int r,i;

   for(i=0; i<10; i++) {
      r = rand() % NUM;
      printf("%d\n",r);
   }
}
```