

1. Explique a diferença entre ordenação interna e externa.

A ordenação interna é usada quando o arquivo cabe na memória RAM, sendo feita a leitura de todos os dados, para serem ordenados direto na RAM e depois registrados no arquivo original. Dessa forma é utilizada para arquivos que cabem na memória RAM. Além disso, na ordenação interna, qualquer registro pode ser imediatamente acessado não importando a ordem.

A ordenação externa é usada quando o arquivo é usado quando o arquivo é maior que o espaço da memória RAM, dessa forma é criado subarquivos a partir do original para a ordenação. Dessa forma é utilizado para ordenar arquivos que não cabem na memória RAM. Além disso, na método de ordenação externa, os registros são acessados de forma sequencial, em grandes blocos e portanto não podemos acessar imediatamente qualquer registro.

2. Faça uma pesquisa sobre a estabilidade de um método de ordenação.

Um método de ordenação é estável se a ordem relativa dos itens com chaves iguais não se altera durante a ordenação, dessa forma elementos iguais vão vir na ordem que aparecerem primeiro. Dessa forma:

3a, 2, 3b, 5, 4, 1

Será ordenado como:

1,2,3a,3b,4,5

3. Crie uma função que implemente o método Shell Sort.

```
void shellSort(int *vet, int qtd){
    int h,aux;
    for(h = 1; h < qtd; h = 3*h+1);
    h = h/3;

    for(; h >=1; h/=3){
        for(int i = h; i < qtd; ++i){
            int j = i;
            for (; vet[j] < vet[j-h] && j >= h; j-=h){
                aux = vet[j];
                vet[j] = vet[j-h];
                vet[j-h] = aux;
            }
        }
    }
}
```

4. Na função main, crie um vetor e o preencha com valores aleatórios. Em sequência, teste a função criada no exercício anterior.

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

```

void shellSort(int *vet, int qtd){
    int h,aux;
    for(h = 1; h < qtd; h = 3*h+1);
    h = h/3;

    for(; h >=1; h/=3){
        for(int i = h; i < qtd; ++i){
            int j = i;
            for (; vet[j] < vet[j-h] && j >= h; j-=h){
                aux = vet[i];
                vet[j] = vet[j-h];
                vet[j-h] = aux;
            }
        }
    }
}

int main() {
    setlocale(LC_ALL, "Portuguese");
    int tam = 45,vetShellS[tam];
    srand(time(NULL));
    for(int i = 0; i < tam; ++i){
        vet[i] = rand()%50;
    }

    printf("\nVetor Original Insertion Sort: \n");
    for(int i = 0; i < tam; ++i){
        printf(" %d |", vet[i]);
    }
    printf("\n");
    printf("Vetor Ordenado Insertion Sort: \n");
    shellSort(vetShellS,tam);
    for(int i = 0; i < tam; ++i){
        printf(" %d |", vet[i]);
    }
    printf("\n");
}

```