Exercício 1: Escreva um programa para mostrar o tempo necessário para o algoritmo Selection Sort ordenar um vetor de 50 mil inteiros carregado com a função RAND()

da stdlib.h.

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main (void){

int tamanho = 50000;

int vet[tamanho], aux, inte, exte;

clock\_t tempoInic, tempoFim, tempoDecorrido;

for(inte=0; inte<tamanho;inte++){

vet[inte]=rand();

}

tempoInic=clock();

//Inicia o relógio

for(exte=0;exte<tamanho-1;exte++){

for(inte=exte+1;inte<tamanho;inte++){

if(vet[exte]>vet[inte]){

aux=vet[exte];

vet[exte]=vet[inte];

vet[inte]=aux;

}

}

}

//Consulta o relógio e mostra o tempo decorrido.

tempoFim=clock();

tempoDecorrido=tempoFim-tempoInic;

printf("Durac=%fms\n",(double)tempoDecorrido/(CLOCKS\_PER\_SEC/1000));

return 0;

}

Exercício 2: Escreva um programa em C para mostrar o tempo necessário para o algoritmo BUBBLE ordenar um vetor de 50 mil inteiros.

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define tam 50000

int main(void) {

int vet[tam], troca;

clock\_t ti, tf, td;

register int pos, aux;

for(pos=0;pos<tam;pos++){

vet[pos]=rand();

}

ti=clock();

troca=1;

while(troca=-1){

troca=0;

for(pos=0;pos<tam-1;pos++){

if(vet[pos]>vet[pos+1]){

aux=vet[pos];

vet[pos]=vet[pos+1];

vet[pos+1]=aux;

troca=1;

}

}

}

tf=clock();

td=tf-ti;

printf("Durac=%.1f\n",(double)td/(CLOCKS\_PER\_SEC));

return 0;

}