

***Asignatura:***

Programación de sistemas

***Sustentado por:***

Chelsea Massiell Suazo García – 61911443

**Dirigido:**

Ing. Kevin Cruz

***Asignación:***

Examen I

***Campus:***

Ceutec Sede Norte

***Fecha:***

20/06/2021

**MÉTODO DE LA BURBUJA**

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#define TAM 100

void imprimeCB(int \*CB) {

int i;

for(i = 0; i < TAM-1; i++) {

printf( "%d, ", CB[i]);

}

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

}

int main() {

int CB[TAM];

int e, i, auxiliar;

srand((unsigned int)time(NULL));

for(e = 0; e < TAM; e++)

CB[e] = (int)(rand() % 100);

printf( "Antes de ordenar\---------------------\n");

imprimeCB(CB);

for(e = 0; e < TAM; e++)

for(i = 0; i < TAM-1-e; i++)

if(CB[i] > CB[i+1]) {

auxiliar = CB[i+1];

CB[i+1] = CB[i];

CB[i] = auxiliar;

}

printf( "\nDespués de ordenar\n------------------\n");

imprimeCB(CB);

}

**BURBUJA MEJORADA**

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#define TAM 100

void imprimeCB(int \*CB) {

int i;

for(i = 0; i < TAM-1; i++) {

printf( "%d, ", CB[i]);

}

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

}

int main() {

int CB[TAM];

int e, i, auxiliar,intercambio;

srand((unsigned int)time(NULL));

for(e = 0; e < TAM; e++)

CB[e] = (int)(rand() % 100);

printf( "Antes de ordenar\n--------------\n");

imprimeCB(CB);

for(e = 0; e < TAM; e++){

intercambio = 0;

for(i = 0; i < TAM-1-e; i++){

if(CB[i] > CB[i+1]) {

auxiliar = CB[i+1];

CB[i+1] = CB[i];

CB[i] = auxiliar;

intercambio =1;

}

}

if (intercambio==0){

printf( "\nPara en la iteración %d\n",e);

break;

}

}

printf( "\nDespués de ordenar\n---------------\n");

imprimeCB(CB);

}

**MÉTODO DE SELECCIÓN**

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#define TAM 100

void imprimeCB(int \*CB) {

int i;

for(i = 0; i < TAM-1; i++) {

printf( "%d, ", CB[i]);

}

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

}

int main() {

int CB[TAM];

int e,i,PosMenor,aux;

srand((unsigned int)time(NULL));

for(e = 0; e < TAM; e++)

CB[e] = (int)(rand() % 100);

printf( "Antes de ordenar\n-----------------------\n");

imprimeCB(CB);

for (e=0; e<(TAM-1) ; e++) {

PosMenor=e;

for (i=e+1;i<TAM;i++)

if (CB[i]<CB[PosMenor])

PosMenor=i;

aux=CB[e];

CB[e]=CB[PosMenor];

CB[PosMenor]=aux;

}

printf( "\nDespués de ordenar\n--------------------\n");

imprimeCB(CB);

}

**MÉTODO DE INSERCIÓN**

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#define TAM 100

void imprimeCB(int \*CB) {

int i;

for(i = 0; i < TAM-1; i++) {

printf( "%d, ", CB[i]);

}

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

}

int main() {

int CB[TAM];

int e,i,k,temp;

srand((unsigned int)time(NULL));

for(e = 0; e < TAM; e++)

CB[e] = (int)(rand() % 100);

printf( "Antes de ordenar\n-----------------------------\n");

imprimeCB(CB);

for (e=1;e<TAM;e++){

temp=CB[e];

i=0;

while (CB[i]<=temp)

i++;

if (i<e)

{

for (k=e;k>i;k--)

CB[k]=CB[k-1];

CB[i]=temp;

}

}

printf( "\nDespués de ordenar\n------------------------\n");

imprimeCB(CB);

}

**QUICKSORT**

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#define TAM 100

void quickSort( int[], int, int);

int partition( int[], int, int);

void imprimeCB(int \*CB) {

int i;

for(i = 0; i < TAM-1; i++) {

printf( "%d, ", CB[i]);

}

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

}

int main() {

int CB[TAM];

int e;

srand((unsigned int)time(NULL));

for(e = 0; e < TAM; e++)

CB[e] = (int)(rand() % 100);

printf( "Antes de ordenar\n----------------------\n");

imprimeCB(CB);

quickSort( CB, 0, TAM-1);

printf( "\nDespués de ordenar\n--------------------\n");

imprimeCB(CB);

}

void quickSort( int CB[], int izquierda, int derecha){

int indice\_pivote;

if( izquierda < derecha ) {

indice\_pivote = partition( CB, izquierda, derecha);

quickSort( CB, izquierda, indice\_pivote-1);

quickSort( CB, indice\_pivote+1, derecha);

}

}

int partition( int CB[], int izquierda, int derecha) {

int pivote, i, j, tmp;

pivote = CB[izquierda];

i = izquierda; j = derecha;

while( 1){

while( CB[i] <= pivote && i <= derecha ) ++i;

while( CB[j] > pivote ) --j;

if( i >= j ) break;

tmp = CB[i]; CB[i] = CB[j]; CB[j] = tmp;

}

tmp = CB[izquierda]; CB[izquierda] = CB[j]; CB[j] = tmp;

return j;

}