**BUBBLESORT**

#include <stdio.h>

#define SIZE 9

int A[SIZE] = {24,13,26,1,2,27,38,15,10};

void print\_array(int \*, int);

void swap(int, int);

void bsort(void);

int main(void)

{

printf("Original: \n");

print\_array(A, SIZE);

bsort();

printf("\nSorted: \n");

print\_array(A, SIZE);

return 0;

}

void bsort()

{

int i, j;

for(i = 0; i < SIZE; i++)

{

for(j = SIZE - 1; j > 0; j--)

if(A[j] < A[j - 1])

swap(j, j - 1);

print\_array(A, SIZE);

}

}

void swap(int i, int j)

{

int temp;

temp = A[i];

A[i] = A[j];

A[j] = temp;

}

void print\_array(int \* X, int)

{

for(int i=0; i<SIZE; i++)

printf("%d \t", X[i]);

printf("\n");

}

**INSERTIONSORT**

#include <stdio.h>

#define SIZE 9

int A[SIZE] = {24,13,26,1,2,27,38,15,10};

void print\_array(int \*, int);

void isort(void);

int main(void)

{

printf("Original: \n");

print\_array(A, SIZE);

isort();

printf("\nSorted: \n");

print\_array(A, SIZE);

return 0;

}

void isort()

{

int i, j;

int temp;

for(i = 0; i < SIZE; i++) {

// find insertion spot

j = i; temp = A[i];

while(A[j - 1] > temp && j > 0) {

A[j] = A[j - 1];

j--;

}

A[j] = temp;

print\_array(A, SIZE);

}

}

void print\_array(int \* X, int)

{

for(int i=0; i<SIZE; i++)

printf("%d \t", X[i]);

printf("\n");

}

**SELECTIONSORT**

#include <stdio.h>

#define SIZE 9

int A[SIZE] = {24,13,26,1,2,27,38,15,10};

void print\_array(int \*, int);

void ssort(void);

void swap(int i, int j);

int main(void)

{

printf("Original: \n");

print\_array(A, SIZE);

ssort();

printf("\nSorted: \n");

print\_array(A, SIZE);

return 0;

}

void ssort()

{

int i, j;

int m;

i = SIZE - 1;

while(i >= 0) {

// find biggest unsorted element

j = 0;

m = 0;

while(j <= i) {

if(A[j] > A[m])

m = j;

j++;

}

// put biggest at end

swap(i, m);

i--;

print\_array(A, SIZE);

}

}

void swap(int i, int j)

{

int temp;

temp = A[i];

A[i] = A[j];

A[j] = temp;

}

void print\_array(int \* X, int)

{

for(int i=0; i<SIZE; i++)

printf("%d \t", X[i]);

printf("\n");

}