Aim: write a program in c to perform selection sort

Algorithm:

1. Find the minimum element in the unsorted portion of the list.
2. Swap the minimum element with the first element of the unsorted portion.
3. Move the boundary of the unsorted portion one element to the right.
4. Repeat the steps above until the unsorted portion is empty.

Source Code:

#include <stdio.h>

*void* swap(*int* \*x, *int* \*y)

{

*int* temp = \*x;

    \*x = \*y;

    \*y = temp;

}

*void* selectionSort(*int* arr[], *int* n)

{

*int* i, j, min\_idx;

    for (i = 0; i < n-1; i++)

    {

        min\_idx = i;

        for (j = i+1; j < n; j++)

        if (arr[j] < arr[min\_idx])

            min\_idx = j;

        swap(&arr[min\_idx], &arr[i]);

    }

}

*void* printArray(*int* arr[], *int* size)

{

*int* i;

    for (i=0; i < size; i++)

        printf("%d ", arr[i]);

    printf("\n");

}

*int* main()

{

*int* arr[] = {64, 25, 12, 22, 11};

*int* n = sizeof(arr)/sizeof(arr[0]);

    selectionSort(arr, n);

    printf("Sorted array: \n");

    printArray(arr, n);

    return 0;

}

Output:

Sorted array:

11 12 22 25 64