**Name: M.Rayyan**

**Roll No: S008**

*LINEAR SEARCH FUNCTION*

#include <iostream>

using namespace std;

void getArray(int arr[], int SIZE)

{

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

{

cout << "Input Value: ";

cin >> arr[i];

}

cout << endl;

}

void linearSearch(int arr[], int SIZE)

{

int key;

cout << "Input Search No: ";

cin >> key;

int found = -1;

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

{

if(key == arr[i])

{

found = i;

break;

}

}

if(found != -1)

cout << key << " found at index " << found << endl;

else

cout << key << " not found " << endl;

}

int main()

{

const int SIZE = 5;

int arr[SIZE];

getArray(arr, SIZE);

linearSearch(arr, SIZE);

return 0;

}

*BUBBLE SORT + BINARY SEARCH FUNCTION*

#include <iostream>

using namespace std;

void getArray(int arr[], int SIZE)

{

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

{

cout << "Input Value: ";

cin >> arr[i];

}

cout << endl;

}

void bubbleSort(int arr[], int SIZE)

{

for(int i = 0; i < SIZE-1; i++)

{

for(int j = 0; j < SIZE-i-1; j++)

{

if(arr[j+1] < arr[j])

swap(arr[j], arr[j+1]);

}

}

}

void showArray(int arr[], int SIZE)

{

cout << "ARRAY AFTER SORT " << endl;

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

{

cout << "Value at index " << i << " is " << arr[i] << endl;

}

cout << endl;

}

void binarySearch(int arr[], int SIZE)

{

int key;

cout << "Input Search No: ";

cin >> key;

int found = -1;

int high = SIZE - 1;

int low = 0;

for(int i = low; i <= high; i++)

{

int middle = low + (high - low) / 2;

if(key < arr[middle])

high = middle - 1;

else if(key > arr[middle])

low = middle + 1;

else

{

found = middle;

break;

}

}

if(found != -1)

cout << key << " found at index " << found << endl;

else

cout << key << " not found." << endl;

}

int main()

{

const int SIZE = 5;

int arr[SIZE];

getArray(arr, SIZE);

bubbleSort(arr, SIZE);

showArray(arr, SIZE);

binarySearch(arr, SIZE);

return 0;

}

*SELECTION SORT+ BINARY SEARCH FUNTION*

#include <iostream>

using namespace std;

void getArray(int arr[], int SIZE)

{

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

{

cout << "Input Value: ";

cin >> arr[i];

}

cout << endl;

}

void selectionSort(int arr[], int SIZE)

{

for(int i = 0; i < SIZE-1; i++)

{

int min = i;

for(int j = i+1; j < SIZE; j++)

{

if(arr[j] < arr[min])

min = j;

}

swap(arr[min], arr[i]);

}

}

void showArray(int arr[], int SIZE)

{

cout << "ARRAY AFTER SORT " << endl;

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

{

cout << "Value at index " << i << " is " << arr[i] << endl;

}

cout << endl;

}

void binarySearch(int arr[], int SIZE)

{

int key;

cout << "Input Search No: ";

cin >> key;

int found = -1;

int high = SIZE - 1;

int low = 0;

for(int i = low; i <= high; i++)

{

int middle = low + (high - low) / 2;

if(key < arr[middle])

high = middle - 1;

else if(key > arr[middle])

low = middle + 1;

else

{

found = middle;

break;

}

}

if(found != -1)

cout << key << " found at index " << found << endl;

else

cout << key << " not found." << endl;

}

int main()

{

const int SIZE = 5;

int arr[SIZE];

getArray(arr, SIZE);

selectionSort(arr, SIZE);

showArray(arr, SIZE);

binarySearch(arr, SIZE);

return 0;

}

*MAX/MIN 2ND MAX/MIN FUNCTION*

#include <iostream>

using namespace std;

void getArray(int arr[], int SIZE)

{

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

{

cout << "Input Value: ";

cin >> arr[i];

}

cout << endl;

}

int maxValue(int arr[], int SIZE)

{

int max1 = arr[0];

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

{

if(arr[i] > max1)

max1 = arr[i];

}

return max1;

}

int minValue(int arr[], int SIZE)

{

int min1 = arr[0];

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

{

if(arr[i] < min1)

min1 = arr[i];

}

return min1;

}

void secondMaxValue(int arr2[], int SIZE)

{

int max1 = arr2[0];

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

{

if(arr2[i] > max1)

max1 = arr2[i];

}

cout << "Second Max: " << max1 << endl;

}

void secondMinValue(int arr2[], int SIZE)

{

int min1 = arr2[0];

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

{

if(arr2[i] < min1)

min1 = arr2[i];

}

cout << "Second Min: " << min1 << endl;

}

int main()

{

const int SIZE = 5;

int arr[SIZE];

getArray(arr, SIZE);

int max = maxValue(arr, SIZE);

cout << "Max value: " << max << endl;

int min = minValue(arr, SIZE);

cout << "Min value: " << min << endl;

int arr2[SIZE];

int k = 0;

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

{

if((arr[i] < max) && (arr[i] > min))

arr2[k++] = arr[i];

}

secondMaxValue(arr2, k);

secondMinValue(arr2, k);

return 0;

}

*SUM/AVG FUNCTION*

#include <iostream>

using namespace std;

void getArray(int ar[], int SIZE)

{

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

{

cout << "Input value: ";

cin >> ar[i];

}

}

float sumArray(int arr[], int size)

{

float sum1 = 0;

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

{

sum1 = sum1 + arr[i];

}

return sum1;

// cout << "\nSum is " << sum << endl; if use void like done avg

}

void avgArray(int ar[], int size, float sum1)

{

float avg;

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

{

avg = sum1 / size;

}

cout << "\nAvg is " << avg << endl;

}

int main()

{

const int SIZE = 5;

int arr[SIZE];

getArray(arr, SIZE); // only name of array ie arr not arr[SIZE]

float sum = sumArray(arr, SIZE);

cout << "\nSum is " << sum << endl;

avgArray(arr, SIZE, sum);

return 0;

}