C++ Lab Record

# Lab - 1

File: search.cpp

## Algorithm:

[Algorithm generation failed]

## Code:

#include <stdio.h>  
#include "headerfile.h"  
  
void linear(int arr[], int num, int value)  
{  
 int found = 0;  
  
 for (int i = 0; i < num; i++)  
 {  
 if (arr[i] == value)  
 {  
 printf("Found\n");  
 found = 1;  
 break;  
 }  
 else  
 {  
 found = 0;  
 }  
 }  
 if (found == 0)  
 {  
 printf("Not found\n");  
 }  
}  
  
void binary(int arr[], int num, int value)  
{  
 int start = 0, end = num - 1, mid;  
 while (start <= end)  
 {  
 mid = (start + end) / 2; // Calculate mid each time  
 if (arr[mid] == value)  
 {  
 printf("Number found\n");  
 return; // Exit the function if the number is found  
 }  
 else if (arr[mid] > value)  
 {  
 end = mid - 1; // Search the left half  
 }  
 else  
 {  
 start = mid + 1; // Search the right half  
 }  
 }  
 printf("Number not found\n"); // Number not found if loop exits  
}  
  
int main()  
{  
 int choice;  
 int num;  
 int value;  
  
 printf("Enter the number of numbers to be entered\n");  
  
 scanf("%d", &num);  
  
 int arr[num];  
  
 printf("Enter the numbers\n");  
  
 for (int i = 0; i < num; i++)  
 {  
 scanf("%d", &arr[i]);  
 }  
  
 printf("\n");  
  
 printf("1. Linear Search\n2. Binary search\n");  
 scanf("%d", &choice);  
  
 printf("Enter the number to search\n");  
 scanf("%d", &value);  
  
 switch (choice)  
 {  
 case 1:  
 linear(arr, num, value);  
 break;  
 case 2:  
 insertion(arr, num);  
 binary(arr, num, value);  
 break;  
 }  
}

## Output:

Enter the number of numbers to be entered  
Enter the numbers  
  
1. Linear Search  
2. Binary search  
Enter the number to search

File: sorting.cpp

## Algorithm:

[Algorithm generation failed]

## Code:

// Menu driven program for sorting  
#include <stdio.h>  
  
// insertion sort  
void insertion(int arr[], int num)  
{  
 for (int i = 0; i < num; i++)  
 {  
 int j = i - 1;  
 int temp = arr[i];  
  
 while (j >= 0 && arr[j] > temp)  
 {  
 arr[j + 1] = arr[j];  
 j--;  
 }  
  
 arr[j + 1] = temp;  
 }  
}  
  
// Bubble Sort  
void bubble(int arr[], int num)  
{  
 int temp1;  
 for (int i = 0; i < num; i++)  
 {  
 for (int j = 0; j < num - i - 1; j++)  
 {  
 if (arr[j] > arr[j + 1])  
 {  
 temp1 = arr[j];  
 arr[j] = arr[j + 1];  
 arr[j + 1] = temp1;  
 }  
 }  
 }  
}  
  
// Selection Sort  
void selection(int arr[], int num)  
{  
 int temp2;  
 for (int i = 0; i < num - 1; i++)  
 {  
 for (int j = i + 1; j < num; j++)  
 {  
 if (arr[j]<arr[i])  
 {  
 temp2 = arr[j];  
 arr[j] = arr[i];  
 arr[i] = temp2;  
 }  
 }  
 }  
}  
  
int main()  
{  
 int num;  
 int choice;  
  
 printf("Enter the number of numbers to be entered\n");  
  
 scanf("%d", &num);  
  
 int arr[num];  
  
 printf("Enter the numbers\n");  
  
 for (int i = 0; i < num; i++)  
 {  
 scanf("%d", &arr[i]);  
 }  
  
 printf("\n");  
  
  
 printf("1. Insertion Sort\n2. Bubble Sort\n3. Selection Sort\n4. Exit\n");  
  
 scanf("%d", &choice);  
  
 switch (choice)  
 {  
 case 1:  
 insertion(arr, num);  
 break;  
 case 2:  
 bubble(arr, num);  
 break;  
 case 3:  
 selection(arr, num);  
 break;  
 case 4:  
 break;  
 }  
   
 for (int i = 0; i < num; i++)  
 {  
 printf("%d", arr[i]);  
 }  
}

## Output:

Enter the number of numbers to be entered  
Enter the numbers  
  
1. Insertion Sort  
2. Bubble Sort  
3. Selection Sort  
4. Exit