Assignment – 1

Q1. Implementation of Iterative Linear Search.

#include <stdio.h>  
#include <stdlib.h>  
  
int\* create\_array(int);  
int linear\_search(int\*, int, int);  
void display\_array(int\*, int);  
  
int main() {  
 int size, key;  
 printf("Enter the size of the array: ");  
 scanf("%d", &size);  
 int\* array = create\_array(size);  
 display\_array(array, size);  
 printf("Enter the key to search in the array: ");  
 scanf("%d", &key);  
 int pos = linear\_search(array, size, key);  
  
 if (pos == -1) {  
 printf("The element %d is not found in the array\n", key);  
 } else {  
 printf("The element %d found at index %d in the array\n", key, pos);  
 }  
 free(array);  
 return 0;  
}  
  
int\* create\_array(int size) {  
 int \*array = (int\*)malloc(sizeof(int) \* size), i;  
 printf("Enter %d array elements: ", size);  
  
 for (i = 0; i < size; i++) {  
 scanf("%d", &array[i]);  
 }  
 return array;  
}  
  
int linear\_search(int\* array, int size, int key) {  
 int i;  
  
 for (i = 0; i < size; i++) {  
 if (array[i] == key) {  
 return i;  
 }  
 }  
 return -1;  
}  
  
void display\_array(int\* array, int size) {  
 int i;  
 printf("The array elements are: ");  
  
 for (i = 0; i < size; i++) {  
 printf("%d ", array[i]);  
 }  
 printf("\n");  
}

Output:

Enter the size of the array: 6  
Enter 6 array elements: 19 2 24 8 10 7  
The array elements are: 19 2 24 8 10 7   
Enter the key to search in the array: 24  
The element 24 found at index 2 in the array

Q2. Implementation of Recursive Linear Search.

#include <stdio.h>  
#include <stdlib.h>  
  
int\* create\_array(int);  
int linear\_search(int\*, int, int);  
void display\_array(int\*, int);  
  
int main() {  
 int size, key;  
 printf("Enter the size of the array: ");  
 scanf("%d", &size);  
 int\* array = create\_array(size);  
 display\_array(array, size);  
 printf("Enter the key to search in the array: ");  
 scanf("%d", &key);  
 int pos = linear\_search(array, size, key);  
  
 if (pos == size) {  
 printf("The element %d is not found in the array\n", key);  
 } else {  
 printf("The element %d found at index %d in the array\n", key, pos);  
 }  
 free(array);  
 return 0;  
}  
  
int\* create\_array(int size) {  
 int \*array = (int\*)malloc(sizeof(int) \* size), i;  
 printf("Enter %d array elements: ", size);  
  
 for (i = 0; i < size; i++) {  
 scanf("%d", &array[i]);  
 }  
 return array;  
}  
  
int linear\_search(int\* array, int size, int key) {  
 if (size == 0 || array[0] == key) {  
 return 0;  
 }  
 return 1 + linear\_search(array + 1, size - 1, key);  
}  
  
void display\_array(int\* array, int size) {  
 int i;  
 printf("The array elements are: ");  
  
 for (i = 0; i < size; i++) {  
 printf("%d ", array[i]);  
 }  
 printf("\n");  
}

Output:

Enter the size of the array: 6  
Enter 6 array elements: 19 2 24 8 10 7  
The array elements are: 19 2 24 8 10 7   
Enter the key to search in the array: 24  
The element 24 found at index 2 in the array

Q3. Implementation of Iterative Binary Search.

#include <stdio.h>  
#include <stdlib.h>  
  
int\* create\_array(int);  
int binary\_search(int\*, int, int);  
void display\_array(int\*, int);  
  
int main() {  
 int size, key;  
 printf("Enter the size of the array: ");  
 scanf("%d", &size);  
 int\* array = create\_array(size);  
 display\_array(array, size);  
 printf("Enter the key to search in the array: ");  
 scanf("%d", &key);  
 int pos = binary\_search(array, size, key);  
  
 if (pos == -1) {  
 printf("The element %d is not found in the array\n", key);  
 } else {  
 printf("The element %d found at index %d in the array\n", key, pos);  
 }  
 free(array);  
 return 0;  
}  
  
int\* create\_array(int size) {  
 int \*array = (int\*)malloc(sizeof(int) \* size), i;  
 printf("Enter %d array elements: ", size);  
  
 for (i = 0; i < size; i++) {  
 scanf("%d", &array[i]);  
 }  
 return array;  
}  
  
int binary\_search(int\* array, int size, int key) {  
 int low = 0, high = size - 1, mid;  
  
 while (low <= high) {  
 mid = (low + high) / 2;  
  
 if (array[mid] < key) {  
 low = mid + 1;  
 } else if (array[mid] > key) {  
 high = mid - 1;  
 } else {  
 return mid;  
 }  
 }  
 return -1;  
}  
  
void display\_array(int\* array, int size) {  
 int i;  
 printf("The array elements are: ");  
  
 for (i = 0; i < size; i++) {  
 printf("%d ", array[i]);  
 }  
 printf("\n");  
}

Output:

Enter the size of the array: 6  
Enter 6 array elements: 2 7 8 10 19 24  
The array elements are: 2 7 8 10 19 24   
Enter the key to search in the array: 24  
The element 24 found at index 5 in the array

Q4. Implementation of Recursive Binary Search.

#include <stdio.h>  
#include <stdlib.h>  
  
int\* create\_array(int);  
int binary\_search(int\*, int, int, int);  
void display\_array(int\*, int);  
  
int main() {  
 int size, key;  
 printf("Enter the size of the array: ");  
 scanf("%d", &size);  
 int\* array = create\_array(size);  
 display\_array(array, size);  
 printf("Enter the key to search in the array: ");  
 scanf("%d", &key);  
 int pos = binary\_search(array, 0, size - 1, key);  
  
 if (pos == -1) {  
 printf("The element %d is not found in the array\n", key);  
 } else {  
 printf("The element %d found at index %d in the array\n", key, pos);  
 }  
 free(array);  
 return 0;  
}  
  
int\* create\_array(int size) {  
 int \*array = (int\*)malloc(sizeof(int) \* size), i;  
 printf("Enter %d array elements: ", size);  
  
 for (i = 0; i < size; i++) {  
 scanf("%d", &array[i]);  
 }  
 return array;  
}  
  
int binary\_search(int\* array, int low, int high, int key) {  
 if (low <= high) {  
 int mid = (low + high) / 2;  
  
 if (array[mid] < key) {  
 return binary\_search(array, mid + 1, high, key);  
 }  
 if (array[mid] > key) {  
 return binary\_search(array, low, mid - 1, key);  
 }  
 return mid;  
 }  
 return -1;  
}  
  
void display\_array(int\* array, int size) {  
 int i;  
 printf("The array elements are: ");  
  
 for (i = 0; i < size; i++) {  
 printf("%d ", array[i]);  
 }  
 printf("\n");  
}

Output:

Enter the size of the array: 6  
Enter 6 array elements: 2 7 8 10 19 24  
The array elements are: 2 7 8 10 19 24   
Enter the key to search in the array: 24  
The element 24 found at index 5 in the array