

Output:

Enter the number of elements in the array: 5

1

2

3

4

5

Enter element to search: 3

Element found at index 2!

Lab Sheet 1

1. Given an array of 'n' integers and a key element, write a C program to search the element using linear search.

Source Code:

```
#include <stdio.h>
```

```
int main() {
    int n, i; key;
    printf("Enter the number of elements in
the array: ");
    scanf("%d", &n);
    int arr[n];
    for(i=0; i<n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter element to search: ");
    scanf("%d", &key);
    for(i=0; i<n; i++) {
        if(arr[i] == key) {
            printf("Element found at index %d\n", i);
            return 0;
        }
    }
}
```

Collected

{

{

printf("Element not found. \n");

return 0;

{

Output

Enter the number of elements in the array: 5

1

2

3

4

5

Enter element to search: 1
Element found at index 0!

2. Given an array of integers implement binary search to find the position of a given key
Source Code:

```
#include <stdio.h>
```

```
int main() {
    int n, i, key;
    printf("Enter the number of elements in the
array: ");
    scanf("%d", &n);
    int arr[n];
    for(i=0; i<n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter element to search: ");
    scanf("%d", &key);
    int low = 0, high = n, mid;
    do {
        mid = (low + high) / 2;
        if (arr[mid] == key) low = mid + 1;
        else if (arr[mid] > key) high = mid - 1;
        else break;
    } while (arr[mid] != key);
    if (low <= high)
}
```



```
if (arr[mid] == key) {
```

```
    printf ("Element found at index %d! \n", mid);
```

```
    return 0;
```

```
}
```

```
printf ("Element not found. \n");
```

```
return 0;
```

```
}
```

Output:

Enter the number of elements in the array : 5

5

4

3

2

1

Which algorithm do you want to sort with?

1. Bubble Sort
2. Selection Sort
3. Insertion Sort

3

Sorting elements using insertion sort(ascending):

1

2

3

4

5

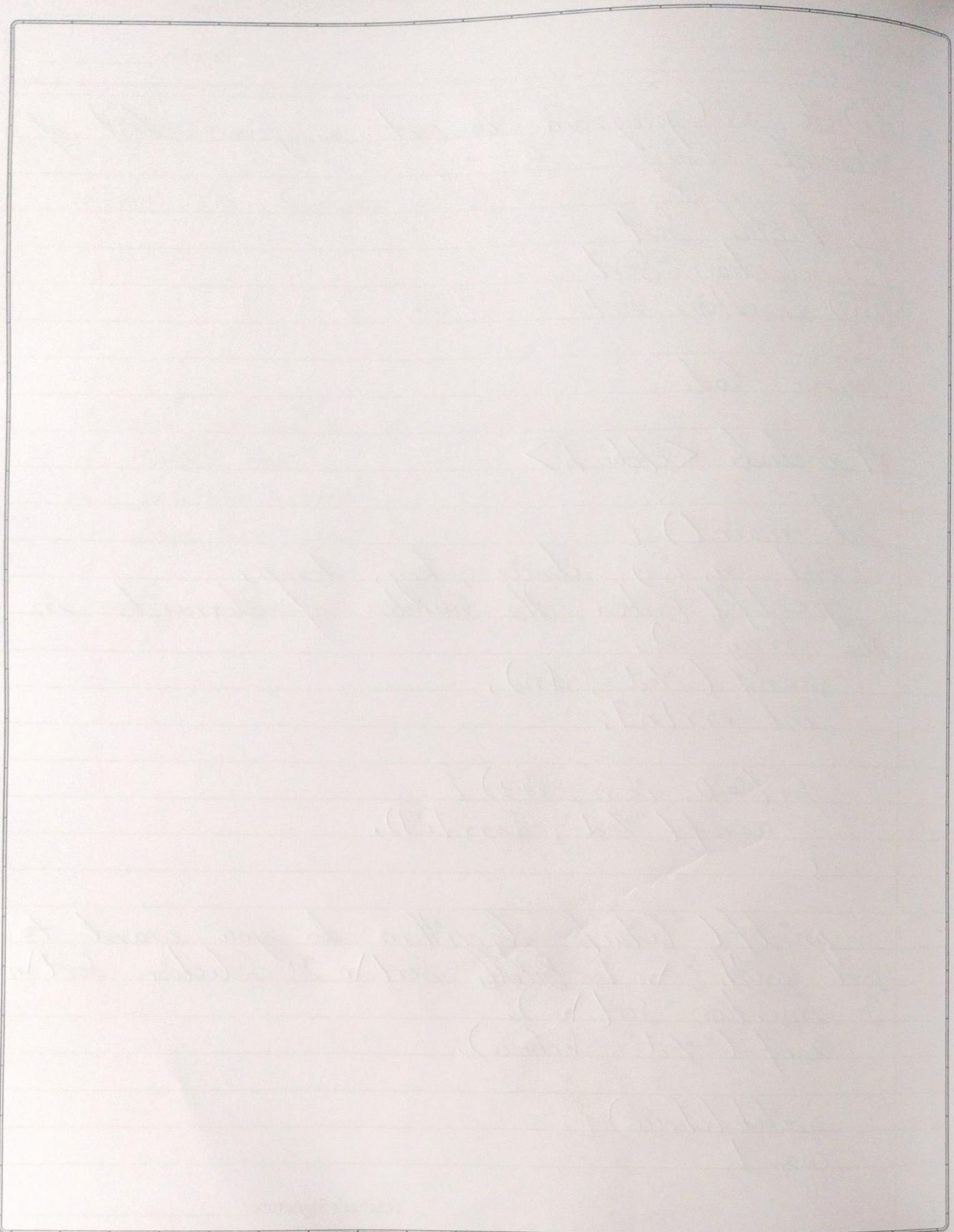
3. Write a C program to sort a given list of elements using:

- (i) bubble sort
- (ii) Selection Sort
- (iii) Insertion sort

Source Code :

```
#include <stdio.h>
```

```
int main() {
    int n, i, j; choice, key, temp;
    printf("Enter the number of elements in
the array : ");
    scanf("%d", &n);
    int arr[n];
    for(i=0; i<n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Which algorithm do you want to
sort with? \n 1. bubble sort \n 2. Selection sort \n
3. Insertion sort \n");
    scanf("%d", &choice);
    switch(choice) {
        case 1:
```



// ===== BUBBLE SORT =====

`printf("Sorting elements using bubble sort
(ascending):\n")`

```
for (int i=0; i<n-1; i++) {
    for (int j=0; j<n-1; j++) {
        if (arr[j] > arr[j+1]) {
            temp = arr[j+1];
            arr[j+1] = arr[j];
            arr[j] = temp;
        }
    }
}
```

```
for (int i=0; i<n; i++) {
    printf("%d\n", arr[i]);
}
```

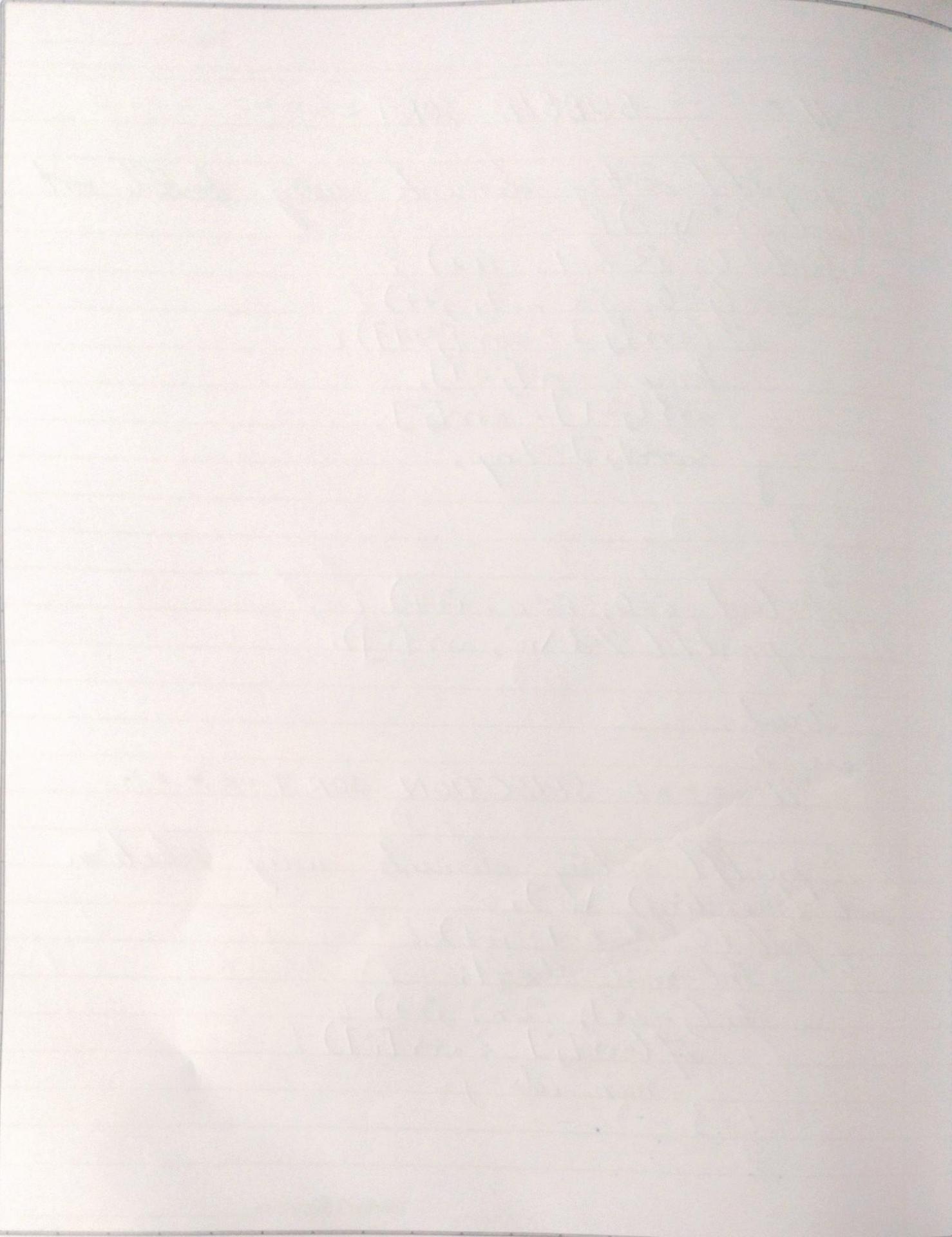
`break;`

`case 2:`

// ===== SELECTION SORT =====

`printf("Sorting elements using selection
sort (ascending):\n")`

```
for (int i=0; i<n-1; i++) {
    int min_idx = i;
    for (int j=i+1; j<n; j++) {
        if (arr[j] < arr[min_idx]) {
            min_idx = j;
        }
    }
}
```



```

temp = arr[i];
arr[i] = arr[min_idx];
arr[min_idx] = temp;
}
for(int i=0; i<n; i++) {
    printf ("%d\n", arr[i]);
}
break;
case 3:
// ===== INSERTION SORT =====

```

```

printf ("Sorting elements using insertion sort
(ascending):\n");
for(int i=1; i<n; i++) {
    key = arr[i];
    j = i-1;
    while (j >= 0 && key < arr[j]) {
        arr[j+1] = arr[j];
        j--;
    }
    arr[j+1] = key;
}
for(int i=0; i<n; i++) {
    printf ("%d\n", arr[i]);
}
break;
default:
    printf ("Invalid choice!\n");
}
return 0;

```