



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

Title:

Array Operations

Problem Statement

Perform various operations on array such as 1. Create 2. Display, 3. Sort, 4. Search, 5. Modify, 6. Insert, 7. Delete. A. Without Pointers B. using pointer (Beyond Syllabus)

Programmer Name: Arpan Agrawal

Batch: E6

1) Array manipulation without Pointer

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

```
void insertionSort(int arr[], int n);  
void selectionSort(int arr[], int n);  
void bubbleSort(int arr[], int n);  
void mergeSort(int arr[], int left, int right);  
void merge(int arr[], int left, int mid, int right);  
int linearSearch(int arr[], int n, int x);  
int binarySearch(int arr[], int left, int right, int x);  
void modify(int arr[], int n);  
void insert(int arr[], int n);  
void delete(int arr[], int n);
```

```
int main() {  
    int choice;  
    int sortChoice, searchChoice;  
    int arr[100];  
    int n, i, x, result;  
  
    printf("Enter the number of elements: ");  
    scanf("%d", &n);  
  
    for(i = 0; i < n; i++) {  
        printf("Enter element %d: ", i + 1);  
        scanf("%d", &arr[i]);  
    }  
  
    do {
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

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

printf("\nMenu:");
printf("\n 1. Sorting");
printf("\n 2. Searching");
printf("\n 3. Insert");
printf("\n 4. Modify");
printf("\n 5. Delete");
printf("\n 6. Exit");
printf("\nEnter choice to perform: ");
scanf("%d", &choice);

switch (choice) {
    case 1:
        printf("\nSort Options:");
        printf("\n 1. Insertion Sort");
        printf("\n 2. Selection Sort");
        printf("\n 3. Bubble Sort");
        printf("\n 4. Merge Sort");
        printf("\nEnter sorting choice: ");
        scanf("%d", &sortChoice);

        switch (sortChoice) {
            case 1:
                insertionSort(arr, n);
                break;
            case 2:
                selectionSort(arr, n);
                break;
            case 3:
                bubbleSort(arr, n);
                break;
            case 4:
                mergeSort(arr, 0, n - 1);
                break;
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

default:

```
printf("Invalid sort choice.\n");
```

```
}
```

```
break;
```

case 2:

```
printf("\nSearch Options:");
```

```
printf("\n 1. Linear Search");
```

```
printf("\n 2. Binary Search");
```

```
printf("\nEnter searching choice: ");
```

```
scanf("%d", &searchChoice);
```

```
switch (searchChoice) {
```

```
case 1:
```

```
printf("\n Enter the number to be find :");
```

```
scanf("%d", &x);
```

```
result = linearSearch(arr, n, x);
```

```
if (result != -1)
```

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

```
else
```

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

```
break;
```

```
case 2:
```

```
printf("\n Enter the number to be find :");
```

```
scanf("%d", &x);
```

```
insertionSort(arr, n);
```

```
result = binarySearch(arr, 0, n - 1, x);
```

```
if (result != -1)
```

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

```
else
```

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

```
break;
```

```
default:
```

```
printf("Invalid search choice.\n");
```

```
}
```

```
break;
```

case 3:



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
        insert(arr, &n);
        break;
    case 4:
        modify(arr, n);
        break;
    case 5:
        delete(arr, &n);
        break;
    case 6:
        printf("Exiting...\n");
        break;
    default:
        printf("Invalid choice.\n");
    }
} while (choice != 6);

return 0;
}

// Insertion sort function
void insertionSort(int arr[], int n) {
    for (int i = 1; i < n; i++) {
        int key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;

        // Display array after each pass
        printf("Pass %d: ", i);
        for (int k = 0; k < n; k++)
            printf("%d ", arr[k]);
        printf("\n");
    }
    printf("\nFinal sorted array: ");
    for (int i = 0; i < n; i++)
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
        printf("%d ", arr[i]);
    printf("\n");
}

// Selection sort function
void selectionSort(int arr[], int n) {
    for (int i = 0; i < n-1; i++) {
        int minIndex = i;
        for (int j = i+1; j < n; j++) {
            if (arr[j] < arr[minIndex]) {
                minIndex = j;
            }
        }
        int temp = arr[minIndex];
        arr[minIndex] = arr[i];
        arr[i] = temp;

        // Display array after each pass
        printf("Pass %d: ", i + 1);
        for (int k = 0; k < n; k++)
            printf("%d ", arr[k]);
        printf("\n");
    }
    printf("\nFinal sorted array: ");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

// Bubble sort function
void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n-1; i++) {
        int swapped = 0;
        for (int j = 0; j < n-i-1; j++) {
            if (arr[j] > arr[j+1]) {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
        swapped = 1;
    }
}

// Display array after each pass
printf("Pass %d: ", i + 1);
for (int k = 0; k < n; k++)
    printf("%d ", arr[k]);
printf("\n");

if (!swapped) {
    break;
}
}
printf("\nFinal sorted array: ");
for (int i = 0; i < n; i++)
    printf("%d ", arr[i]);
printf("\n");
}

//merge is the function which is called in merge sort function
void merge(int arr[], int left, int mid, int right) {
    int n1 = mid - left + 1;
    int n2 = right - mid;
    int L[n1], R[n2];

    for (int i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (int j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];

    int i = 0, j = 0, k = left;

    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        }
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
        } else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }

    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

void mergeSort(int arr[], int left, int right) {
    if (left < right) {
        int mid = left + (right - left) / 2;
        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);
        merge(arr, left, mid, right);

        printf("Merging: ");
        for (int i = left; i <= right; i++)
            printf("%d ", arr[i]);
        printf("\n");
    }
}

// Linear search function
int linearSearch(int arr[], int n, int x) {
    for (int i = 0; i < n; i++) {
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
        if (arr[i] == x)
            return i;
    }
    return -1;
}

// Binary search function
int binarySearch(int arr[], int left, int right, int x) {
    if (right >= left) {
        int mid = left + (right - left) / 2;
        if (arr[mid] == x)
            return mid;
        if (arr[mid] > x)
            return binarySearch(arr, left, mid - 1, x);
        return binarySearch(arr, mid + 1, right, x);
    }
    return -1;
}

// To modify the array
void modify(int arr[], int n) {
    int index, newValue;

    printf("Enter the index of the element to modify (0 to %d): ", n-1);
    scanf("%d", &index);

    if (index < 0 || index >= n) {
        printf("Invalid index!\n");
        return;
    }

    printf("Enter the new value: ");
    scanf("%d", &newValue);

    arr[index] = newValue;
    printf("Element at index %d has been updated to %d.\n", index, newValue);
}
```




PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

// To insert an element in the array

```
void insert(int arr[], int *n) {  
    int index, value;
```

```
    if (*n >= 100) {  
        printf("Array is full. Cannot insert more elements.\n");  
        return;  
    }
```

```
    printf("Enter the index where you want to insert the new element (0 to %d): ", *n);  
    scanf("%d", &index);
```

```
    if (index < 0 || index > *n) {  
        printf("Invalid index!\n");  
        return;  
    }
```

```
    printf("Enter the value to insert: ");  
    scanf("%d", &value);
```

```
    for (int i = *n; i > index; i--) {  
        arr[i] = arr[i - 1];  
    }
```

```
    arr[index] = value;  
    (*n)++;  
    printf("Element inserted at index %d.\n", index);  
}
```

// To delete an element from the array

```
void delete(int arr[], int *n) {  
    int index;
```

```
    if (*n <= 0) {  
        printf("Array is empty. Nothing to delete.\n");  
        return;  
    }
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
printf("Enter the index of the element to delete (0 to %d): ", *n - 1);  
scanf("%d", &index);
```

```
if (index < 0 || index >= *n) {  
    printf("Invalid index!\n");  
    return;  
}
```

```
for (int i = index; i < *n - 1; i++) {  
    arr[i] = arr[i + 1];  
}
```

```
(*n)--;  
printf("Element deleted from index %d.\n", index);  
}
```

2) Array manipulation with Pointer

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

```
void insertionSort(int *arr, int n);  
void selectionSort(int *arr, int n);  
void bubbleSort(int *arr, int n);  
void mergeSort(int *arr, int left, int right);  
void merge(int *arr, int left, int mid, int right);  
int linearSearch(int *arr, int n, int x);  
int binarySearch(int *arr, int left, int right, int x);  
void modify(int *arr, int n);  
void insert(int *arr, int *n);  
void delete(int *arr, int *n);
```

```
int main() {  
    int choice;  
    int sortChoice, searchChoice;  
    int arr[100];  
    int n, i, x, result;
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
printf("Enter the number of elements: ");  
scanf("%d", &n);
```

```
for (i = 0; i < n; i++) {  
    printf("Enter element %d: ", i + 1);  
    scanf("%d", &arr[i]);  
}
```

```
do {  
    printf("\nArray elements are: ");  
    for (i = 0; i < n; i++) {  
        printf("%d ", arr[i]);  
    }  
}
```

```
printf("\nMenu:");  
printf("\n 1. Sorting");  
printf("\n 2. Searching");  
printf("\n 3. Insert");  
printf("\n 4. Modify");  
printf("\n 5. Delete");  
printf("\n 6. Exit");  
printf("\nEnter choice to perform: ");  
scanf("%d", &choice);
```

```
switch (choice) {  
    case 1:  
        printf("\nSort Options:");  
        printf("\n 1. Insertion Sort");  
        printf("\n 2. Selection Sort");  
        printf("\n 3. Bubble Sort");  
        printf("\n 4. Merge Sort");  
        printf("\nEnter sorting choice: ");  
        scanf("%d", &sortChoice);
```

```
        switch (sortChoice) {  
            case 1:  
                insertionSort(arr, n);  
                break;
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
case 2:
    selectionSort(arr, n);
    break;
case 3:
    bubbleSort(arr, n);
    break;
case 4:
    mergeSort(arr, 0, n - 1);
    break;
default:
    printf("Invalid sort choice.\n");
}
break;

case 2:
    printf("\nSearch Options:");
    printf("\n 1. Linear Search");
    printf("\n 2. Binary Search");
    printf("\nEnter searching choice: ");
    scanf("%d", &searchChoice);

switch (searchChoice) {
case 1:
    printf("\n Enter the number to be found: ");
    scanf("%d", &x);
    result = linearSearch(arr, n, x);
    if (result != -1)
        printf("Element found at index %d\n", result);
    else
        printf("Element not found\n");
    break;
case 2:
    printf("\n Enter the number to be found: ");
    scanf("%d", &x);
    insertionSort(arr, n);
    result = binarySearch(arr, 0, n - 1, x);
    if (result != -1)
        printf("Element found at index %d\n", result);
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
        else
            printf("Element not found\n");
        break;
    default:
        printf("Invalid search choice.\n");
    }
    break;

    case 3:
        insert(arr, &n);
        break;
    case 4:
        modify(arr, n);
        break;
    case 5:
        delete(arr, &n);
        break;
    case 6:
        printf("Exiting...\n");
        break;
    default:
        printf("Invalid choice.\n");
    }
} while (choice != 6);

return 0;
}

void insertionSort(int *arr, int n) {
    for (int i = 1; i < n; i++) {
        int key = *(arr + i);
        int j = i - 1;
        while (j >= 0 && *(arr + j) > key) {
            *(arr + j + 1) = *(arr + j);
            j--;
        }
        *(arr + j + 1) = key;
    }
}
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
// Display array after each pass
printf("Pass %d: ", i);
for (int k = 0; k < n; k++)
    printf("%d ", *(arr + k));
printf("\n");
}
printf("\nFinal sorted array: ");
for (int i = 0; i < n; i++)
    printf("%d ", *(arr + i));
printf("\n");
}

void selectionSort(int *arr, int n) {
    for (int i = 0; i < n - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < n; j++) {
            if (*(arr + j) < *(arr + minIndex)) {
                minIndex = j;
            }
        }
        int temp = *(arr + minIndex);
        *(arr + minIndex) = *(arr + i);
        *(arr + i) = temp;

        // Display array after each pass
        printf("Pass %d: ", i + 1);
        for (int k = 0; k < n; k++)
            printf("%d ", *(arr + k));
        printf("\n");
    }
    printf("\nFinal sorted array: ");
    for (int i = 0; i < n; i++)
        printf("%d ", *(arr + i));
    printf("\n");
}

void bubbleSort(int *arr, int n) {
    for (int i = 0; i < n - 1; i++) {
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

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

// Display array after each pass
printf("Pass %d: ", i + 1);
for (int k = 0; k < n; k++)
    printf("%d ", *(arr + k));
printf("\n");

if (!swapped) {
    break;
}

printf("\nFinal sorted array: ");
for (int i = 0; i < n; i++)
    printf("%d ", *(arr + i));
printf("\n");
}

void merge(int *arr, int left, int mid, int right) {
    int n1 = mid - left + 1;
    int n2 = right - mid;
    int L[n1], R[n2];

    for (int i = 0; i < n1; i++)
        L[i] = *(arr + left + i);
    for (int j = 0; j < n2; j++)
        R[j] = *(arr + mid + 1 + j);

    int i = 0, j = 0, k = left;
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
while (i < n1 && j < n2) {
    if (L[i] <= R[j]) {
        *(arr + k) = L[i];
        i++;
    } else {
        *(arr + k) = R[j];
        j++;
    }
    k++;
}

while (i < n1) {
    *(arr + k) = L[i];
    i++;
    k++;
}

while (j < n2) {
    *(arr + k) = R[j];
    j++;
    k++;
}

void mergeSort(int *arr, int left, int right) {
    if (left < right) {
        int mid = left + (right - left) / 2;
        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);
        merge(arr, left, mid, right);

        printf("Merging: ");
        for (int i = left; i <= right; i++)
            printf("%d ", *(arr + i));
        printf("\n");
    }
}
```




PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
int linearSearch(int *arr, int n, int x) {  
    for (int i = 0; i < n; i++) {  
        if (*(arr + i) == x)  
            return i;  
    }  
    return -1;  
}
```

```
int binarySearch(int *arr, int left, int right, int x) {  
    if (right >= left) {  
        int mid = left + (right - left) / 2;  
        if (*(arr + mid) == x)  
            return mid;  
        if (*(arr + mid) > x)  
            return binarySearch(arr, left, mid - 1, x);  
        return binarySearch(arr, mid + 1, right, x);  
    }  
    return -1;  
}
```

```
void modify(int *arr, int n) {  
    int index, newValue;  
  
    printf("Enter the index of the element to modify (0 to %d): ", n - 1);  
    scanf("%d", &index);  
  
    if (index < 0 || index >= n) {  
        printf("Invalid index!\n");  
        return;  
    }  
  
    printf("Enter the new value: ");  
    scanf("%d", &newValue);  
  
    *(arr + index) = newValue;  
    printf("Element at index %d has been updated to %d.\n", index, newValue);  
}
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
void insert(int *arr, int *n) {
    int index, value;

    if (*n >= 100) {
        printf("Array is full. Cannot insert more elements.\n");
        return;
    }

    printf("Enter the index where you want to insert the new element (0 to %d): ", *n);
    scanf("%d", &index);

    if (index < 0 || index > *n) {
        printf("Invalid index!\n");
        return;
    }

    printf("Enter the value to insert: ");
    scanf("%d", &value);

    for (int i = *n; i > index; i--) {
        *(arr + i) = *(arr + i - 1);
    }

    *(arr + index) = value;
    (*n)++;
    printf("Element inserted at index %d.\n", index);
}

void delete(int *arr, int *n) {
    int index;

    if (*n <= 0) {
        printf("Array is empty. Nothing to delete.\n");
        return;
    }

    printf("Enter the index of the element to delete (0 to %d): ", *n - 1);
    scanf("%d", &index);
```



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

```
if (index < 0 || index >= *n) {  
    printf("Invalid index!\n");  
    return;  
}  
  
for (int i = index; i < *n - 1; i++) {  
    *(arr + i) = *(arr + i + 1);  
}  
  
(*n)--;  
printf("Element deleted from index %d.\n", index);  
}
```

OUTPUT

//SORTING

Enter the number of elements: 5

Enter element 1: 5

Enter element 2: 4

Enter element 3: 3

Enter element 4: 2

Enter element 5: 1

Array elements are: 5 4 3 2 1

Menu:

1. Sorting
2. Searching
3. Insert
4. Modify
5. Delete
6. Exit

Enter choice to perform: 1

Sort Options:



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

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

// Insertion Sort

Enter sorting choice: 1

Pass 1: 4 5 3 2 1

Pass 2: 3 4 5 2 1

Pass 3: 2 3 4 5 1

Pass 4: 1 2 3 4 5

Final sorted array: 1 2 3 4 5

// Selection Sort

Enter sorting choice: 2

Pass 1: 1 4 3 2 5

Pass 2: 1 2 3 4 5

Pass 3: 1 2 3 4 5

Pass 4: 1 2 3 4 5

Final sorted array: 1 2 3 4 5

//Bubble Sort

Enter sorting choice: 3

Pass 1: 4 3 2 1 5

Pass 2: 3 2 1 4 5

Pass 3: 2 1 3 4 5

Pass 4: 1 2 3 4 5

Final sorted array: 1 2 3 4 5

//Merge Sort

Enter sorting choice: 4

Merging: 4 5

Merging: 3 4 5

Merging: 1 2



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

Merging: 1 2 3 4 5

//SEARCHING

Enter the number of elements: 5

Enter element 1: 1

Enter element 2: 2

Enter element 3: 3

Enter element 4: 4

Enter element 5: 5

Array elements are: 1 2 3 4 5

Menu:

1. Sorting
2. Searching
3. Insert
4. Modify
5. Delete
6. Exit

Enter choice to perform: 2

Search Options:

1. Linear Search
2. Binary Search

// Linear Search

Enter searching choice: 1

// Number found

Enter the number to be find :2



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

Element found at index : 1

// Number not found

Enter the number to be find :6

Element not found

// Binary Search

Enter searching choice: 2

//Checking whether the array is sorted or not

Enter the number to be find :3

Pass 1: 1 2 3 4 5

Pass 2: 1 2 3 4 5

Pass 3: 1 2 3 4 5

Pass 4: 1 2 3 4 5

// Number found

Final sorted array: 1 2 3 4 5

Element found at index 2

// Number not found

Enter the number to be find :8

Pass 1: 1 2 3 4 5

Pass 2: 1 2 3 4 5

Pass 3: 1 2 3 4 5

Pass 4: 1 2 3 4 5

Final sorted array: 1 2 3 4 5

Element not found



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

//Inserting a new element to the array

Enter the number of elements: 5

Enter element 1: 5

Enter element 2: 4

Enter element 3: 3

Enter element 4: 2

Enter element 5: 1

Array elements are: 5 4 3 2 1

Menu:

1. Sorting
2. Searching
3. Insert
4. Modify
5. Delete
6. Exit

Enter choice to perform: 3

Enter the index where you want to insert the new element (0 to 5): 4

Enter the value to insert: 34

Element inserted at index 4.

Array elements are: 5 4 3 2 34 1

//Modifying an existing element to the array

Enter choice to perform: 4

Enter the index of the element to modify (0 to 4): 2

Enter the new value: 90

Element at index 2 has been updated to 90.

Array elements are: 5 4 90 2 1



PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043

Department of Electronics & Telecommunication

ASSESSMENT YEAR: 2024-2025

CLASS: SE

SUBJECT: DATA STRUCTURES

EXPT No:

LAB Ref: SE/2024-25/

Starting date:

Roll No: 22203

Submission date:

//Deleting an existing element to the array

Enter choice to perform: 5

Enter the index of the element to delete (0 to 4): 4

Element deleted from index 4.

Array elements are: 5 4 90 2