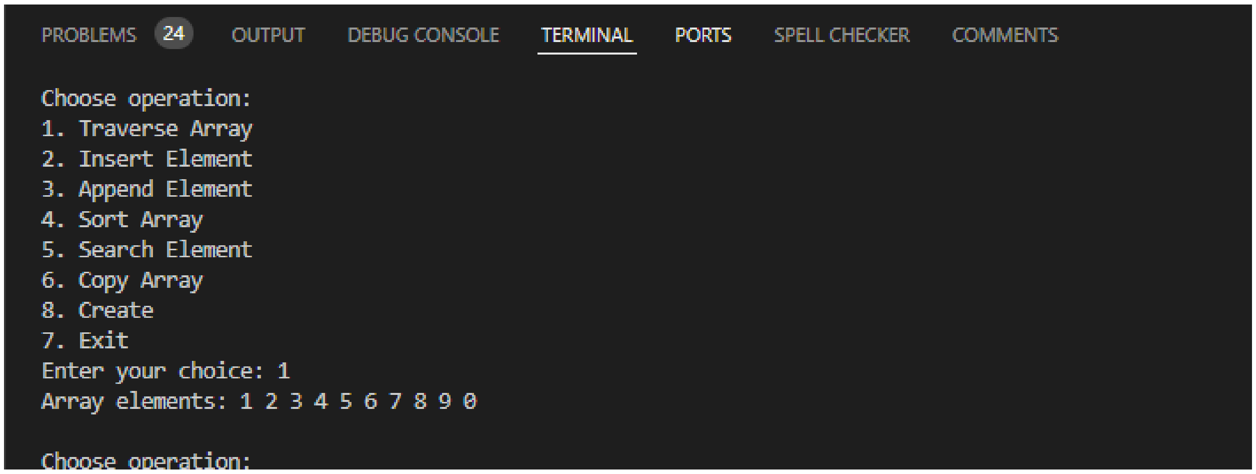
Name: Mithlesh Yeole
Class: B3   Batch: B3
Roll no. : 59

DS Lab practical 1:

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

struct Array {
    int *A;
    int size;
    int length;
};

void create(struct Array *arr, int Max) {
    arr->size = Max;
    arr->A = (int*)malloc(arr->size * sizeof(int));
    arr->length = 0;

    printf("Enter %d elements for the array: \n", arr->size);
    for (int i = 0; i < arr->size; i++) {
        scanf("%d", &arr->A[i]);
    }
    arr->length = arr->size;
}

void append(struct Array *arr, int x) {
    if (arr->length < arr->size) {
        arr->A[arr->length] = x;
        arr->length++;
    } else {
        printf("Array is full. Cannot append.\n");
    }
}

void traverse(struct Array *arr) {
    for (int i = 0; i < arr->length; i++) {
        printf("%d ", arr->A[i]);
    }
    printf("\n");
}
```

```c
void insert(struct Array *arr, int index, int x) {
    if (index >= 0 && index <= arr->length && arr->length < arr->size) {
        for (int i = arr->length - 1; i >= index; i--) {
            arr->A[i + 1] = arr->A[i];
        }
        arr->A[index] = x;
        arr->length++;
    } else {
        printf("Invalid index or array is full.\n");
    }
}
void sortArray(int *arr, int size) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = i + 1; j < size; j++) {
            if (arr[i] > arr[j]) {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

int searchElement(int *arr, int size, int element) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == element) return i;
    }
    return -1;
}
void copyArray(int *arr, int size, int *copyArr) {
    for (int i = 0; i < size; i++) {
        copyArr[i] = arr[i];
    }
}

int main() {
    struct Array arr;
    int maxSize = 10;
    int choice, element, index, found;
    int *copyArr = (int*)malloc(maxSize * sizeof(int));

    create(&arr, maxSize);

    do {
        printf("\nChoose operation: \n");
        printf("1. Traverse Array\n");
        printf("2. Insert Element\n");
        printf("3. Append Element\n");
        printf("4. Sort Array\n");
        printf("5. Search Element\n");
        printf("6. Copy Array\n");
        printf("8. Create\n");
        printf("7. Exit\n");

        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Array elements: ");
                traverse(&arr);
                break;
            case 2:
                printf("Enter element to insert: ");
                scanf("%d", &element);
                printf("Enter index: ");
                scanf("%d", &index);
                insert(&arr, index, element);
                break;
            case 3:
                printf("Enter element to append: ");
                scanf("%d", &element);
                append(&arr, element);
                break;
            case 4:
                sortArray(arr.A, arr.length);
                printf("Array sorted.\n");
                break;
            case 5:
                printf("Enter element to search: ");
                scanf("%d", &element);
                found = searchElement(arr.A, arr.length, element);
                if (found != -1)
                    printf("Element found at index %d\n", found);
                else
                    printf("Element not found\n");
                break;
            case 6:
                copyArray(arr.A, arr.length, copyArr);
                printf("Array copied: ");
                for (int i = 0; i < arr.length; i++) {
                    printf("%d ", copyArr[i]);
                }
                printf("\n");
                break;
            case 7:
                printf("Exiting program.\n");
                break;
            case 8:
            printf("Creating new array...\n");
            create(&arr, maxSize);
            break;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 8);

    return 0;
}
```

Output



```
Choose operation:
1. Traverse Array
2. Insert Element
3. Append Element
4. Sort Array
5. Search Element
6. Copy Array
8. Create
7. Exit
Enter your choice: 1
Array elements: 1 2 3 4 5 6 7 8 9 0

Choose operation:
```