QUICKSORT

Quicksort is a widely used sorting algorithm known for its efficiency and simplicity. It belongs to the divide-and-conquer category of algorithms and works by partitioning an array into smaller subarrays, sorting each subarray, and combining them to produce a sorted array.

Algorithm

```
Partition(A, p, r)
   x = A[r]
   i = p-1
   for j = p to r-1
      if A[j] <= x
         i = i+1
         swap A[i] with A[j]
   swap A[i+1] with A[r]
   return i+1

QuickSort(A, p, r)
    if p<r
        q = Partition(A, p, r)
        QuickSort(A, p, q-1)
        QuickSort(A, q+1, r)
```

IMPLEMENTATION

```c
#include <stdio.h>

void swap(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}


int partition(int arr[], int p, int r) {
    int x = arr[r];
    int i = p - 1;

    for (int j = p; j < r; j++) {
        if (arr[j] < x) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
```

```
    swap(&arr[i + 1], &arr[r]);
    return i + 1;
}

void quicksort(int arr[], int p, int r) {
    if (p < r) {
        int q = partition(arr, p, r);
        quicksort(arr, p, q - 1);
        quicksort(arr, q + 1, r);
    }
}

int main() {
    int n;

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

    int arr[n];

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


    quicksort(arr, 0, n - 1);

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

    return 0;
}
```

OUTPUT

Enter the number of elements: 10
Enter 10 elements:
4 5 2 7 6 11 23 12 9 15

Sorted Array:
2 4 5 6 7 9 11 12 15 23