AMRITA VISHWA VIDYAPEETHAM

DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956

SCHOOL OF COMPUTING

DESIGN AND ANALYSIS OF ALGORITHMS

LAB WORKBOOK

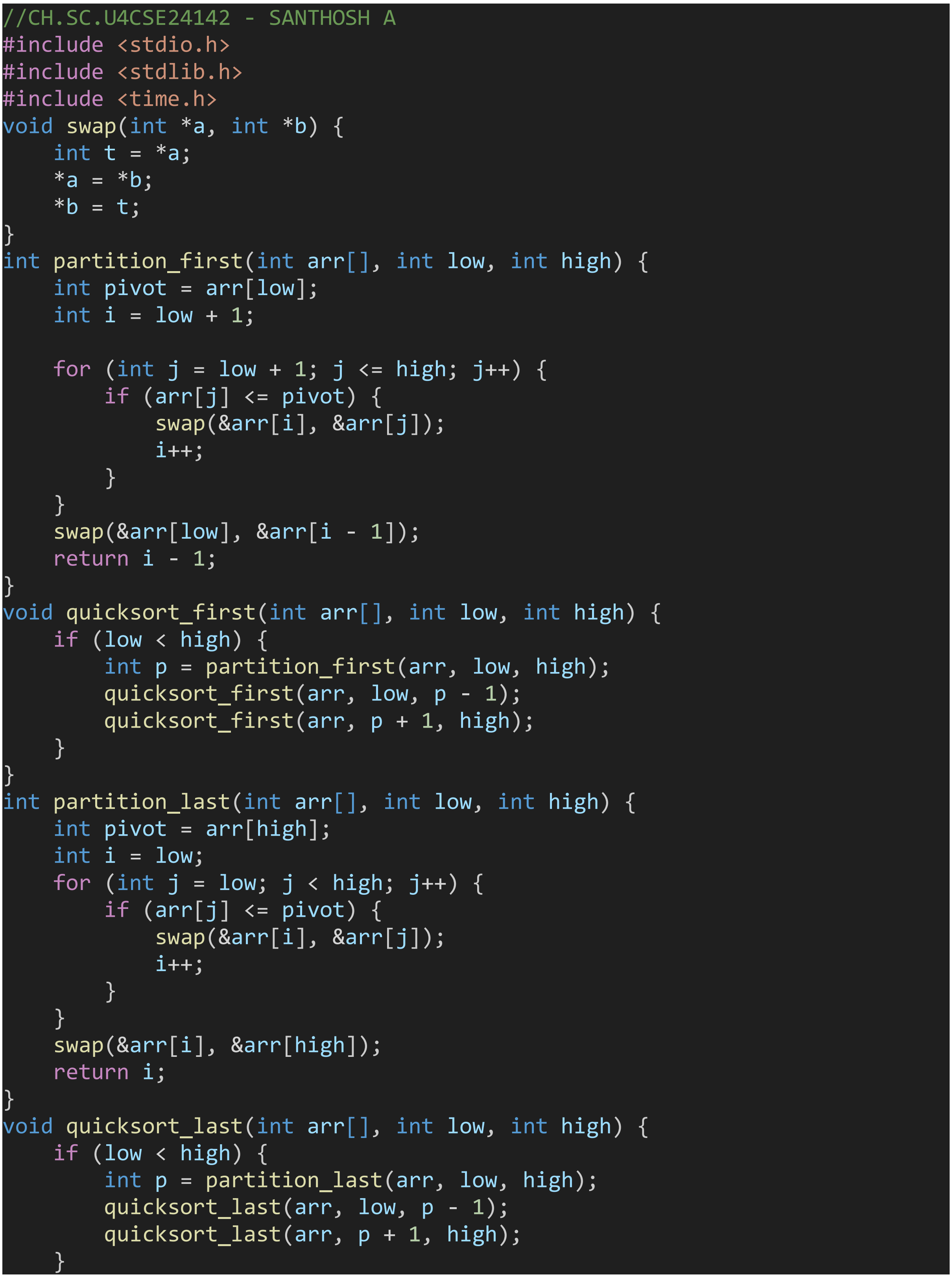
WEEK - 6

NAME	: SANTHOSH A
ROLL NUMBER	: CH.SC.U4CSE24142
CLASS	: CSE-B

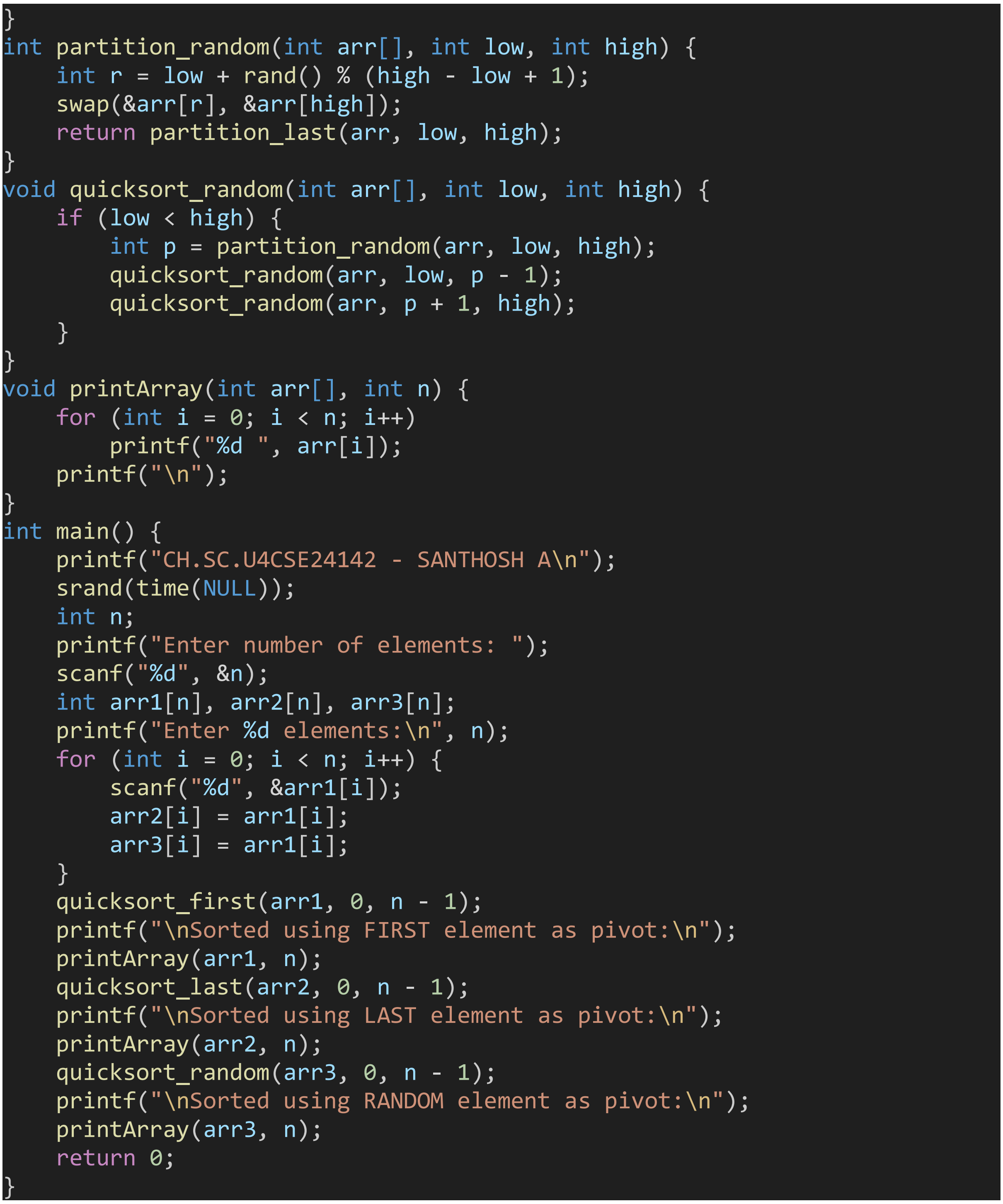
Question 1: Write a program to perform Quick Sort by taking First Element, Last Element and a Random Element as Pivot Element for the given numbers:

157, 110, 147, 122, 149, 151, 111, 141, 112, 123, 133, 117

CODE:

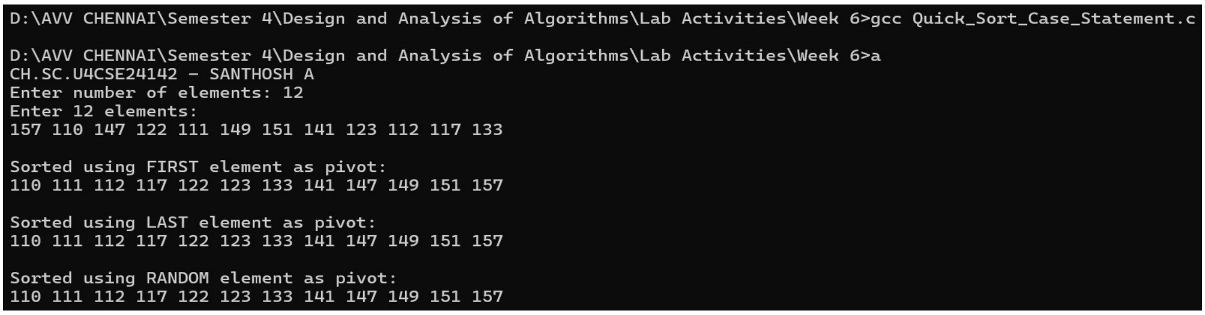


```c
//CH.SC.U4CSE24142 - SANTHOSH A
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}
int partition_first(int arr[], int low, int high) {
    int pivot = arr[low];
    int i = low + 1;

    for (int j = low + 1; j <= high; j++) {
        if (arr[j] <= pivot) {
            swap(&arr[i], &arr[j]);
            i++;
        }
    }
    swap(&arr[low], &arr[i - 1]);
    return i - 1;
}
void quicksort_first(int arr[], int low, int high) {
    if (low < high) {
        int p = partition_first(arr, low, high);
        quicksort_first(arr, low, p - 1);
        quicksort_first(arr, p + 1, high);
    }
}
int partition_last(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = low;
    for (int j = low; j < high; j++) {
        if (arr[j] <= pivot) {
            swap(&arr[i], &arr[j]);
            i++;
        }
    }
    swap(&arr[i], &arr[high]);
    return i;
}
void quicksort_last(int arr[], int low, int high) {
    if (low < high) {
        int p = partition_last(arr, low, high);
        quicksort_last(arr, low, p - 1);
        quicksort_last(arr, p + 1, high);
    }
```



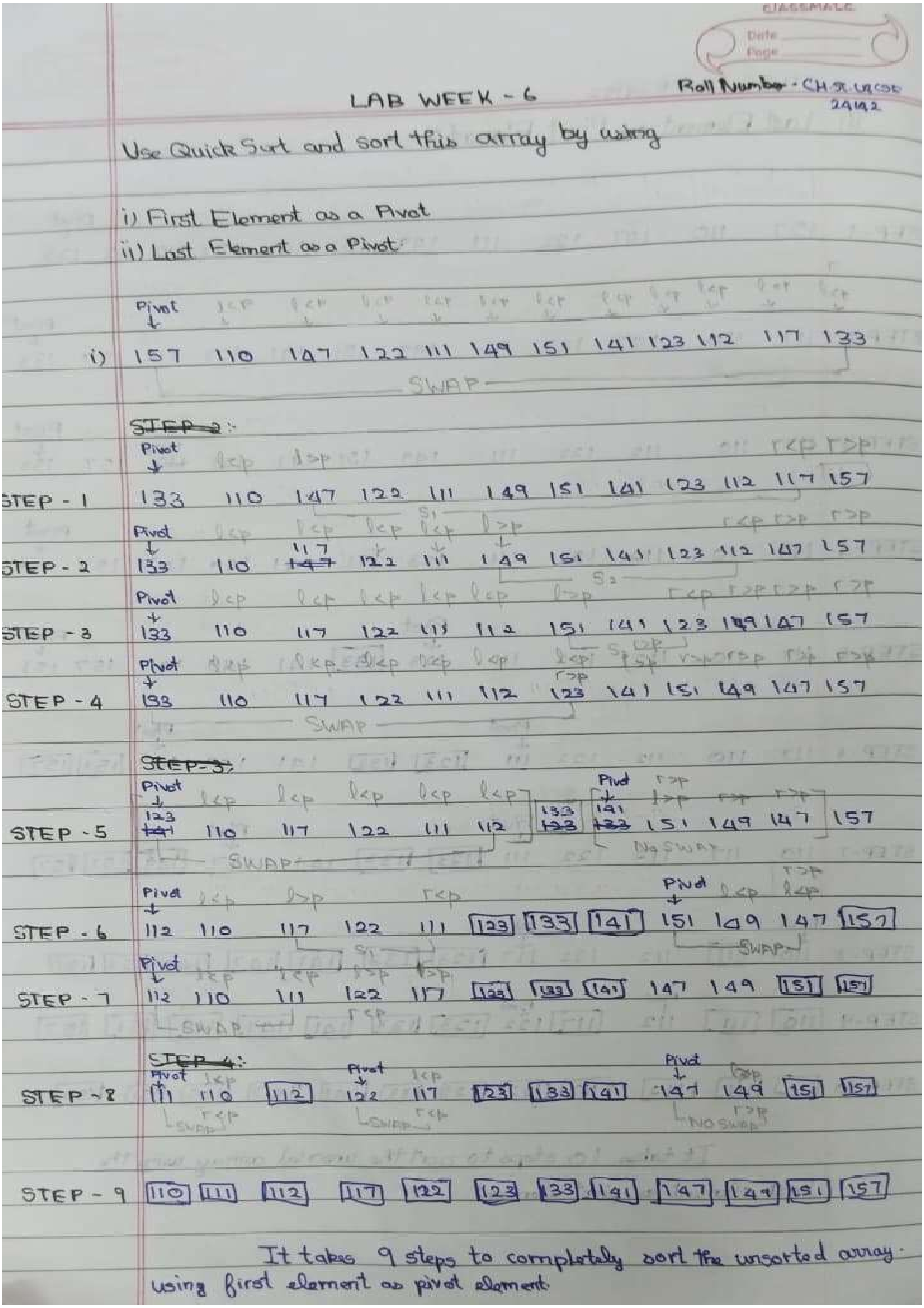
```
}
int partition_random(int arr[], int low, int high) {
    int r = low + rand() % (high - low + 1);
    swap(&arr[r], &arr[high]);
    return partition_last(arr, low, high);
}
void quicksort_random(int arr[], int low, int high) {
    if (low < high) {
        int p = partition_random(arr, low, high);
        quicksort_random(arr, low, p - 1);
        quicksort_random(arr, p + 1, high);
    }
}
void printArray(int arr[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
}
int main() {
    printf("CH.SC.U4CSE24142 - SANTHOSH A\n");
    srand(time(NULL));
    int n;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    int arr1[n], arr2[n], arr3[n];
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr1[i]);
        arr2[i] = arr1[i];
        arr3[i] = arr1[i];
    }
    quicksort_first(arr1, 0, n - 1);
    printf("\nSorted using FIRST element as pivot:\n");
    printArray(arr1, n);
    quicksort_last(arr2, 0, n - 1);
    printf("\nSorted using LAST element as pivot:\n");
    printArray(arr2, n);
    quicksort_random(arr3, 0, n - 1);
    printf("\nSorted using RANDOM element as pivot:\n");
    printArray(arr3, n);
    return 0;
}
```

OUTPUT:



```
D:\AVV CHENNAI\Semester 4\Design and Analysis of Algorithms\Lab Activities\Week 6>gcc Quick_Sort_Case_Statement.c

D:\AVV CHENNAI\Semester 4\Design and Analysis of Algorithms\Lab Activities\Week 6>a
CH.SC.U4CSE24142 - SANTHOSH A
Enter number of elements: 12
Enter 12 elements:
157 110 147 122 111 149 151 141 123 112 117 133

Sorted using FIRST element as pivot:
110 111 112 117 122 123 133 141 147 149 151 157

Sorted using LAST element as pivot:
110 111 112 117 122 123 133 141 147 149 151 157

Sorted using RANDOM element as pivot:
110 111 112 117 122 123 133 141 147 149 151 157
```

WORKING:



Date ________
Page ________

LAB WEEK - 6

Roll Number - CH.SC.U4CSE
24142

Use Quick Sort and sort this array by using

i) First Element as a Pivot
ii) Last Element as a Pivot

Pivot
i) 157 110 147 122 111 149 151 141 123 112 117 133
SWAP

~~STEP 2~~:

STEP - 1 Pivot 133 110 147 122 111 149 151 141 123 112 117 157
S1

STEP - 2 Pivot 133 110 ~~147~~ 117 122 111 149 151 141 123 112 147 157
S2

STEP - 3 Pivot 133 110 117 122 111 112 151 141 123 149 147 157
S3

STEP - 4 Pivot 133 110 117 122 111 112 123 141 151 149 147 157
SWAP

~~STEP-3~~:

STEP - 5 Pivot ~~141~~ 123 110 117 122 111 112 ~~123~~ 133 Pivot ~~133~~ 141 151 149 147 157
SWAP
No SWAP

STEP - 6 Pivot 112 110 117 122 111 [123] [133] [141] Pivot 151 149 147 [157]
S1
SWAP

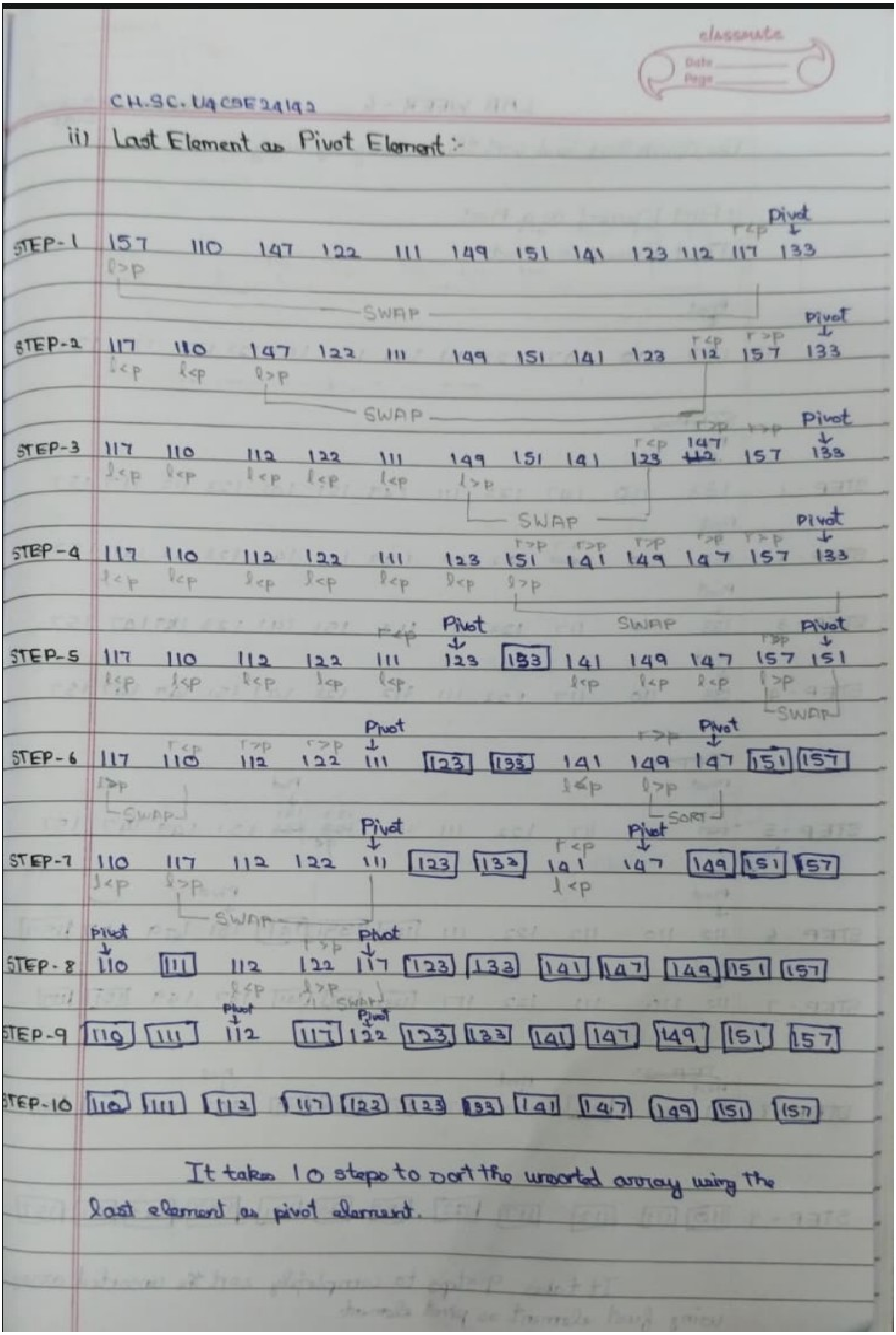
STEP - 7 Pivot 112 110 111 122 117 [123] [133] [141] 147 149 [151] [157]
SWAP

~~STEP 4~~:

STEP - 8 Pivot 111 110 [112] Pivot 122 117 [123] [133] [141] Pivot 147 149 [151] [157]
SWAP SWAP NO SWAP

STEP - 9 [110] [111] [112] [117] [122] [123] [133] [141] [147] [149] [151] [157]

It takes 9 steps to completely sort the unsorted array. using first element as pivot element



CH.SC.U4CSE24142

ii) Last Element as Pivot Element :-

```
                                                                     r<p   Pivot
                                                                            ↓
STEP-1   157   110   147   122   111   149   151   141   123   112   117   133
         l>p
          └─────────────────────────── SWAP ─────────────────────────┘

                                                               r<p   r>p   Pivot
                                                                            ↓
STEP-2   117   110   147   122   111   149   151   141   123   112   157   133
         l<p   l<p   l>p
                      └──────────────── SWAP ───────────────────┘

                                                               r>p   r>p
                                                         r<p   147         Pivot
                                                                            ↓
STEP-3   117   110   112   122   111   149   151   141   123 ~~112~~ 157   133
         l<p   l<p   l<p   l<p   l<p   l>p
                                        └────── SWAP ─────┘

                                             r>p   r>p   r>p   r>p   r>p   Pivot
                                                                            ↓
STEP-4   117   110   112   122   111   123   151   141   149   147   157   133
         l<p   l<p   l<p   l<p   l<p   l<p   l>p
                                              └─────────── SWAP ───────────┘

                                 r<p   Pivot                         r>p   Pivot
                                        ↓                                   ↓
STEP-5   117   110   112   122   111   123  [133]  141   149   147   157   151
         l<p   l<p   l<p   l<p   l<p               l<p   l<p   l<p   l>p
                                                                      └SWAP┘

               r<p   r>p   r>p   Pivot                   r>p   Pivot
                                  ↓                             ↓
STEP-6   117   110   112   122   111  [123] [133]  141   149   147  [151] [157]
         l>p                                       l<p   l>p
          └SWAP┘                                          └SORT┘

                                 Pivot             r<p   Pivot
                                  ↓                       ↓
STEP-7   110   117   112   122   111  [123] [133]  141   147  [149] [151] [157]
         l<p   l>p                                 l<p
                └───── SWAP ─────┘

         Pivot             r>p   Pivot
          ↓                       ↓
STEP-8   110  [111]  112   122   117  [123] [133] [141] [147] [149] [151] [157]
                     l<p   l>p
                            └SWAP┘

                     Pivot       Pivot
                      ↓           ↓
STEP-9  [110] [111]  112  [117]  122  [123] [133] [141] [147] [149] [151] [157]

STEP-10 [110] [111] [112] [117] [122] [123] [133] [141] [147] [149] [151] [157]
```

It takes 10 steps to sort the unsorted array using the last element, as pivot element.

CH.SC.U4CSE24142

iii) Random Element as Pivot Element

STEP-1: 157 110 147 122 111 149 151 141 123 112 117 133

Pivot: 141 ; r<p: 133

SWAP

STEP-2: 133 110 147 122 111 149 151 141 123 112 117 157

Pivot: 141 ; r<p: 117 ; r>p: 157

l<p l<p l>p

SWAP

STEP-3: 133 110 117 122 111 149 151 141 123 112 147 157

Pivot: 141 ; r<p: 112 ; r>p: 147 ; r>p: 157

l<p l<p l<p l<p l<p l>p

SWAP

STEP-4: 133 110 117 122 111 112 151 141 123 149 147 157

Pivot: 141 ; r<p: 123 ; r>p: 149 ; r>p: 147 ; r>p: 157

l<p l<p l<p l<p l<p l<p l>p

SWAP

STEP-5: 133 110 117 122 111 112 ~~151~~ 123 141 151 149 147 157

Pivot: 141 ; r>p: 151 ; r>p: 149 ; r>p: 147 ; r>p: 157

l<p l<p l<p l<p l<p l<p l<p NOSWAP l>p

STEP-6: 133 110 117 122 111 112 123 $\boxed{141}$ 151 149 147 157

Pivot: 111 ; r<p: 110 ; r>p: 117 ; r>p: 122 ; r>p: 112 ; r>p: 123

l>p SWAP

Pivot: 147 ; r>p: 151 ; r>p: 149 ; r>p: 157

l>p SWAP

STEP-7: 110 ~~110~~ 133 117 122 111 ~~133~~ 112 123 $\boxed{141}$ $\boxed{147}$ 149 151 157

Pivot: 111 ; r>p: 133 ; r>p: 117 ; r>p: 122 ; r>p: 112 ; r>p: 123

l<p l>p SWAP

Pivot: 157 ; r<p: 151

l>p l>p NOSWAP

STEP-8: 110 $\boxed{111}$ 117 122 133 112 123 $\boxed{141}$ $\boxed{147}$ 149 151 $\boxed{157}$

Pivot: 110 ; Pivot: 112 ; r>p: 117 ; r>p: 122 ; r>p: 133 ; r>p: 123 ; Pivot: 151

l>p SWAP

STEP-9: $\boxed{110}$ $\boxed{111}$ $\boxed{112}$ 122 133 117 123 $\boxed{141}$ $\boxed{147}$ 149 $\boxed{151}$ $\boxed{157}$

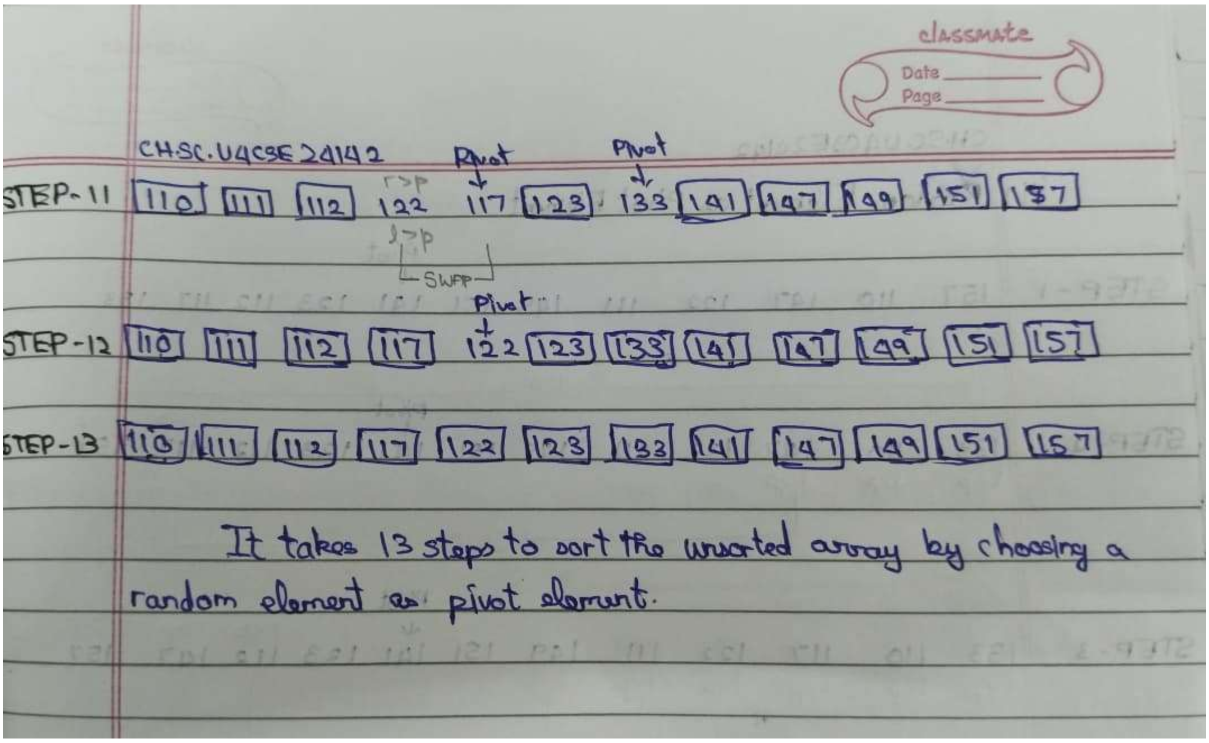
Pivot: 123 ; r<p: 117 ; Pivot: 149

l<p l>p SWAP

STEP-10: $\boxed{110}$ $\boxed{111}$ $\boxed{112}$ 122 117 133 123 $\boxed{141}$ $\boxed{147}$ $\boxed{149}$ $\boxed{151}$ $\boxed{157}$

Pivot: 123 ; r>p: 133

l<p l<p l>p



classmate

Date

Page

CH.SC.U4CSE24142

STEP-11 110 111 112 122 117 123 133 141 147 149 151 157

r>p Pivot Pivot

l>p SWAP

STEP-12 110 111 112 117 122 123 133 141 147 149 151 157

Pivot

STEP-13 110 111 112 117 122 123 133 141 147 149 151 157

It takes 13 steps to sort the unsorted array by choosing a random element as pivot element.

Time Complexity: O(N^2) : It takes O(N^2) time in the worst case because each partition processes all elements while reducing the problem size by only one element at a time.

Space Complexity: O(N) : O(N) space in the worst case because recursive calls can go as deep as the number of elements when partitions are unbalanced.