QUICK SORT

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

void split(int[], int, int);
int partition(int[], int, int);

int main() {
    int a[15000], n, i, j, ch, temp;
    clock_t start, end;

    while (1) {
        printf("1: For manual entry of N value and array elements");
        printf("\n2: To display time taken for sorting number of elements N in the range 500 to 14500");
        printf("\n3: To exit");
        printf("\nEnter your choice: ");
        scanf("%d", &ch);

        switch (ch) {
            case 1:
                printf("Enter the number of elements: ");
                scanf("%d", &n);
                printf("Enter array elements: ");
                for (i = 0; i < n; i++) {
                    scanf("%d", &a[i]);
                }
                start = clock();
                split(a, 0, n - 1);
                end = clock();
                printf("Sorted array is: ");
                for (i = 0; i < n; i++) {
                    printf("%d\t", a[i]);
                }
                printf("Time taken to sort %d numbers is %f Secs\n", n, ((double)(end - start)) / CLOCKS_PER_SEC);
                break;

            case 2:
                n = 500;
                while (n <= 14500) {
                    for (i = 0; i < n; i++) {
                        a[i] = n - i;
                    }
                    start = clock();
                    split(a, 0, n - 1);
                    // Dummy loop to create delay
                    for (j = 0; j < 50000000; j++) { temp = 38 / 600; }
```

```c
                end = clock();
                printf("Time taken to sort %d numbers is %f Secs\n", n, ((double)(end - start)) /
CLOCKS_PER_SEC);
                n += 1000;
            }
            break;

        case 3:
            exit(0);

        default:
            printf("\nInvalid choice! Please try again.\n");
        }
    }
    return 0;
}

void split(int a[], int l, int h) {
    int j;
    if (l < h) {
        j = partition(a,l,h);
        split(a, l, j);
        split(a, j+ 1, h);

    }
}

int partition(int a[], int l, int h) {
    int pivot=a[l];
    int i=l+1;
    int j=h;
    int temp,t;
    do
    {
        while(i<=h && pivot>=a[i])
        {
            i++;
        }
        while(pivot<a[j])
        {
            j--;
        }
        if(i<j)
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;

        }
```

```
    }while(i<j);
    t=a[l];
    a[l]=a[j];
    a[j]=t;
    return j;
}
```
Output

```
1: For manual entry of N value and array elements
2: To display time taken for sorting number of elements N in the range 500 to 14500
3: To exit
Enter your choice: 1
Enter the number of elements: 8
Enter array elements: 22 55 77 11 5 6 3 8
Sorted array is: 3      5      6      8      11     22     55     77      Time taken to sort 8 numbers is 0.000000 Secs
1: For manual entry of N value and array elements
2: To display time taken for sorting number of elements N in the range 500 to 14500
3: To exit
Enter your choice: 2
Time taken to sort 500 numbers is 0.031000 Secs
Time taken to sort 1500 numbers is 0.016000 Secs
Time taken to sort 2500 numbers is 0.016000 Secs
Time taken to sort 3500 numbers is 0.015000 Secs
Time taken to sort 4500 numbers is 0.031000 Secs
Time taken to sort 5500 numbers is 0.047000 Secs
Time taken to sort 6500 numbers is 0.047000 Secs
Time taken to sort 7500 numbers is 0.047000 Secs
Time taken to sort 8500 numbers is 0.063000 Secs
Time taken to sort 9500 numbers is 0.062000 Secs
Time taken to sort 10500 numbers is 0.078000 Secs
Time taken to sort 11500 numbers is 0.094000 Secs
Time taken to sort 12500 numbers is 0.094000 Secs
Time taken to sort 13500 numbers is 0.109000 Secs
Time taken to sort 14500 numbers is 0.110000 Secs
1: For manual entry of N value and array elements
2: To display time taken for sorting number of elements N in the range 500 to 14500
3: To exit
Enter your choice: 3
```

GRAPH

N value	QS time(Secs)
500	0.031
1500	0.016
2500	0.016
3500	0.015
4500	0.031
5500	0.047
6500	0.047
7500	0.047
8500	0.063
9500	0.062
10500	0.078
11500	0.094
12500	0.094
13500	0.109
14500	0.11

QUICK SORT

TIME(SECS) vs N VALUE

QS time(Secs)