

Application Based on Quick Sort

Title : Hospital Queue System

Program Code :

```
#include <stdio.h>

#define SIZE 50

struct Patient {

    int id;

    int priority;

};

void swap(struct Patient *x, struct Patient *y) {

    struct Patient temp = *x;

    *x = *y;

    *y = temp;

}

int partition(struct Patient arr[], int low, int high) {

    int pivot = arr[low].priority;

    int start = low;

    int end = high;

    while (start < end) {

        while (arr[start].priority >= pivot && start < high)
```

```
        start++;

        while (arr[end].priority < pivot)

            end--;

        if (start < end)

            swap(&arr[start], &arr[end]);

    }

    swap(&arr[low], &arr[end]);

    return end;

}

void quick_sort(struct Patient arr[], int low, int high) {

    if (low < high) {

        int loc = partition(arr, low, high);

        quick_sort(arr, low, loc - 1);

        quick_sort(arr, loc + 1, high);

    }

}

int main() {

    struct Patient arr[SIZE];

    int n, i;

    printf("Enter number of patients (max %d): ", SIZE);
```

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

printf("Enter patient IDs and priority levels (1 to 100):\n");

for (i = 0; i < n; i++) {

    printf("Patient %d ID: ", i + 1);

    scanf("%d", &arr[i].id);

    printf("Priority: ");

    scanf("%d", &arr[i].priority);

}

quick_sort(arr, 0, n - 1);

printf("\nPatients sorted by priority (highest first):\n");

printf("ID\tPriority\n");

for (i = 0; i < n; i++) {

    printf("%d\t%d\n", arr[i].id, arr[i].priority);

}

return 0;
```

Output :

C:\Users\USER\Desktop\241410128\Application Quick Sort new.exe

```
Enter number of patients (max 50): 4
Enter patient IDs and priority levels (1 to 100):
Patient 1 ID: 5
Priority: 90
Patient 2 ID: 6
Priority: 85
Patient 3 ID: 7
Priority: 70
Patient 4 ID: 8
Priority: 98
```

```
Patients sorted by priority (highest first):
```

ID	Priority
8	98
5	90
6	85
7	70

```
-----  
Process exited after 57.43 seconds with return value 0  
Press any key to continue . . .
```