DS Assignment

Name – Shrinivas Hatyalikar
Div – CS-B
Roll no – 24

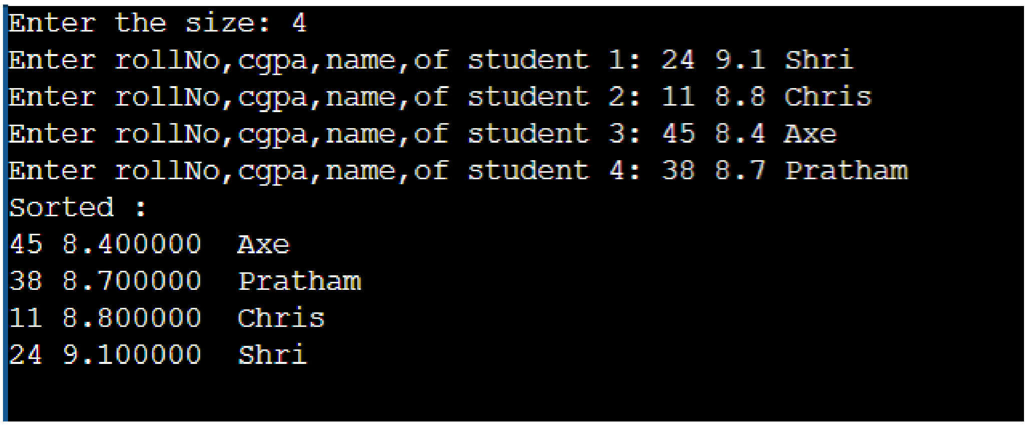
1. Merge sort

Code :

```
#include <stdio.h>
struct student
{
   int rollNo;
   float cgpa;
   char name[20];
};
// int count = 0;
// void swap(struct student a[], int i, int j)
// {
//     struct student temp = a[j];
//     a[j] = a[i];
//     a[i] = temp;
//     count++;
// }
void merge(struct student a[], int low, int k, int high)
{
   struct student result[20];
   int i = low, j = k + 1, l = 0;
   while (i <= k && j <= high)
   {
      if (a[i].cgpa < a[j].cgpa)
      {
         result[l]= a[i];
         i++;
         l++;
      }
      else
      {
         result[l] = a[j];
         j++;
```

```
            l++;
        }
    }
    while (i <= k)
    {
        result[l] = a[i];

        i++;
        l++;
    }
    while (j <= high)
    {
        result[l] = a[j];
        j++;
        l++;
    }
    for (int i = low, j = 0; i <= high; i++, j++)
    {
        a[i] = result[j];
    }
}
void mergesort(struct student a[], int low, int high)
{

    if (low < high)
    {
        int k = (low + high) / 2;
        mergesort(a, low, k);
        mergesort(a, k + 1, high);
        merge(a, low, k, high);
    }
}

int main(int argc, char const *argv[])
{
    int n;
    printf("Enter the size: ");
    scanf("%d", &n);

    struct student s[n];

    for (int i = 0; i < n; i++)
    {
        printf("Enter rollNo,cgpa,name,of student %d: ", i + 1);
        scanf("%d %f %s", &s[i].rollNo, &s[i].cgpa, s[i].name);
    }
    mergesort(s, 0, n - 1);
    printf("Sorted :\n");
    for (int i = 0; i < n; i++)
    {
        printf("%d %f  %s\n", s[i].rollNo, s[i].cgpa, s[i].name);
    }
    // printf("The total number of swaps are %d", count);

    return 0;
}
```

Output :



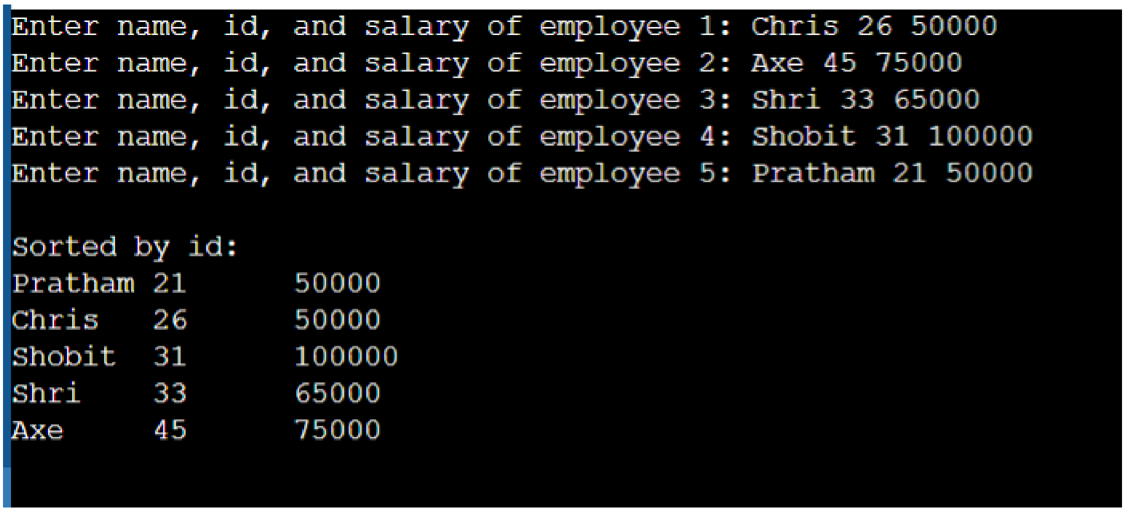
```
Enter the size: 4
Enter rollNo,cgpa,name,of student 1: 24 9.1 Shri
Enter rollNo,cgpa,name,of student 2: 11 8.8 Chris
Enter rollNo,cgpa,name,of student 3: 45 8.4 Axe
Enter rollNo,cgpa,name,of student 4: 38 8.7 Pratham
Sorted :
45 8.400000  Axe
38 8.700000  Pratham
11 8.800000  Chris
24 9.100000  Shri
```

2. Quick sort

Code:

```
#include<stdio.h>

struct employee{
    char name[20];
    int salary;
    int id;
};

void swap(struct employee s[], int i, int j) {
    struct employee temp = s[i];
    s[i] = s[j];
    s[j] = temp;
    // swap++;
}

int partition(struct employee a[], int low, int high)
{
    int pivot = low;
    int i = low;
    int j = high;
    while (i < j)
    {
        while (a[i].id <= a[pivot].id)
        {
            i++;
        }
        while (a[j].id > a[pivot].id)
        {
            j--;
        }

        if (i < j)
        {
            swap(a,i,j);
            // int temp = a[i].id;
            // a[i].id = a[j].id;
            // a[j].id = temp;
        }
    }

    swap(a,pivot,j);
    // int temp = a[pivot].id;
    // a[pivot].id = a[j].id;
    // a[j].id = temp;

    return j;
}

void quicksort(struct employee a[], int low, int high)
{
    int i = low;
    int j = high;
    if (i < j)
    {
        int index = partition(a, low, high);
        quicksort(a, low, index - 1);
        quicksort(a, index + 1, high);
    }
}

int main(){
    int n=5;
    struct employee s[5];
    for(int i=0;i<5;i++){
        printf("Enter name, id, and salary of employee %d: ", i+1);
        scanf("%s %d %d", s[i].name, &s[i].id, &s[i].salary);
    }

    quicksort(s,0,n-1);

    printf("\nSorted by id:\n");
    for (int i = 0; i < 5; i++) {
        printf("%s\t%d\t%d\n", s[i].name, s[i].id, s[i].salary);
    }
    return 0;

}
```

Output:



```
Enter name, id, and salary of employee 1: Chris 26 50000
Enter name, id, and salary of employee 2: Axe 45 75000
Enter name, id, and salary of employee 3: Shri 33 65000
Enter name, id, and salary of employee 4: Shobit 31 100000
Enter name, id, and salary of employee 5: Pratham 21 50000

Sorted by id:
Pratham 21      50000
Chris   26      50000
Shobit  31      100000
Shri    33      65000
Axe     45      75000
```

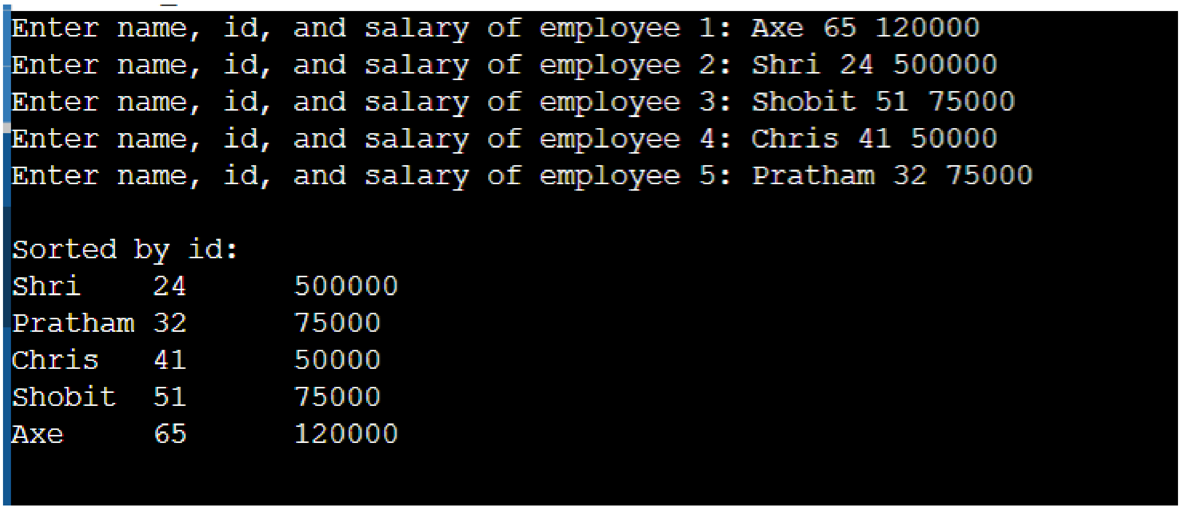
3. Heap sort

Code :

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

```
struct employee{
    char name[20];
    int salary;
    int id;
};


void swap(struct employee s[], int i, int j) {
    struct employee temp = s[i];
    s[i] = s[j];
    s[j] = temp;
}


int heapify(struct employee a[],int N,int parent){
    int largest,left,right;
    largest = parent;
    left = 2*parent+1;
    right =2*parent+2;

    if(left<N && a[left].id>a[largest].id){
        largest=left;
    }

    if(right<N && a[right].id>a[largest].id){
        largest=right;
    }

    if(parent!=largest){
        swap(a,parent,largest);
        // int temp =a[parent];
        // a[parent]=a[largest];
        // a[largest]=temp;
        heapify(a,N,largest);
    }
}

void heapsort(struct employee a[],int N){
    for(int i=N/2-1;i>=0;i--){
        heapify(a,N,i);
    }
    for(int i=N-1;i>=0;i--){
        swap(a,0,i);
        // int temp=a[0];
        // a[0]=a[i];
        // a[i]=temp;
        heapify(a,i,0);
    }
}

int main(){
    struct employee s[5];
    int n=5;
    for(int i=0;i<5;i++){
        printf("Enter name, id, and salary of employee %d: ", i+1);
        scanf("%s %d %d", s[i].name, &s[i].id, &s[i].salary);
    }

    heapsort(s,n);

    printf("\nSorted by id:\n");
    for (int i = 0; i < 5; i++) {
        printf("%s\t%d\t%d\n", s[i].name, s[i].id, s[i].salary);
    }
    return 0;
}
```

Output:



```
Enter name, id, and salary of employee 1: Axe 65 120000
Enter name, id, and salary of employee 2: Shri 24 500000
Enter name, id, and salary of employee 3: Shobit 51 75000
Enter name, id, and salary of employee 4: Chris 41 50000
Enter name, id, and salary of employee 5: Pratham 32 75000

Sorted by id:
Shri    24      500000
Pratham 32      75000
Chris   41      50000
Shobit  51      75000
Axe     65      120000
```