Week-2

Inlab:

1. Insertion sort is partially implemented in the IDE; read the code, complete it and submit to solve the problem.
   Link:https://www.codechef.com/learn/course/college-design-analysis algorithms/CPDAA08/problems/DAA030
   Program:

```
#include <stdio.h>
void Insertion_Sort(int arr[], int n) {
    for(int i = 1; i < n; i++) {
        int j = i;
        int temp = arr[i];
        while(j>0 && arr[j-1] > temp) {
            arr[j] = arr[j-1];
            j--;
        }
        arr[j] = temp;
    }
}

int main() {
    int n;
    scanf("%d", &n);

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

    Insertion_Sort(arr, n);

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

    return 0;
}
```

2. Sort Colors Given an array nums with n objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue. We will use the integers 0, 1, and 2 to represent the color red, white, and blue, respectively. You must solve this problem without using the library's sort function.

Link:https://leetcode.com/problems/sort-colors/?envType=problem-list
v2&envId=sorting
Program:

```
void sortColors(int* nums, int numsSize)
{
   for(int i=0;i<numsSize;i++)
   {
     int min=i;
for(int j=i+1;j<numsSize;j++)
{
        if(nums[min]>nums[j])
            min=j;
    }
      int temp=nums[i];
      nums[i]=nums[min];
      nums[min]=temp;
   }
}
```

3. Sorting
One common task for computers is to sort data. For example, people might want to see all their files on a computer sorted by size. Since sorting is a simple problem with many different possible solutions, it is often used to introduce the study of algorithms. Insertion Sort These challenges will cover Insertion Sort, a simple and intuitive sorting algorithm. We will first start with a nearly sorted list.
Link:https://www.hackerrank.com/challenges/insertionsort1/problem
Program:

```
#include <stdio.h>

void print(int n,int a[]) {
   int i;
   for(i=0; i<n;i++) {
      printf("%d ",a[i]);
   }
   printf("\n");
}
void  insertionsort(int a[],int n);
 int main()
{
int a[20],i,n;
scanf("%d", &n);
for(i=0;i<n;i++)
scanf("%d",&a[i]);
insertionsort(a,n);
}
void insertionsort (int a[],int n)
 {
   int tmp,i,j;
   for (i=1;i<n;i++)
   {
```

```
        j=i-1;
        tmp=a[i];
        while((j>=0)&&(a[j]>tmp))
        {
           a[j+1]=a[j];
           j--;
           print(n,a);
        }
        a[j+1]=tmp;
        print(n,a);
     }
  }
```

Post-lab:

1. Maximum Product of Three Numbers
   Given an integer array nums, find three numbers whose product is maximum and return the maximum product. [Hint: Solve using shell sort] Link: https://leetcode.com/problems/maximum-product-of-three numbers/description/?envType=problem-list-v2&envId=sorting
   Program:

```
int maximumProduct(int* nums, int numsSize) {
    int first_max = INT_MIN;
    int second_max = first_max;
    int third_max = first_max;

    int first_min = INT_MAX;
    int second_min = first_min;

    for (int i = 0; i < numsSize; i++)
    {
        if (first_max <= nums[i])
      {
            third_max = second_max;
            second_max = first_max;
            first_max = nums[i];
        }
else if (second_max <= nums[i])
  {
            third_max = second_max;
            second_max = nums[i];
    }
else if (third_max <= nums[i])
            third_max = nums[i];
        if (first_min >= nums[i])
        {
            second_min = first_min;
            first_min = nums[i];
        }
 else if (second_min >= nums[i])
            second_min = nums[i];
```

```
    }
        int maxi = fmax(first_max * second_max * third_max, first_min * second_min *
    first_max);

        return maxi;
    }
```

2. Assign Cookies Assume you are an awesome parent and want to give your children some cookies. But you should give each child at most one cookie. Each child i has a greed factor g[i], which is the minimum size of a cookie that the child will be content with; and each cookie j has a size s[j]. If s[j] >= g[i], we can assign the cookie j to the child i, and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number. [Hint: Use insertion or shell sort] Link:https://leetcode.com/problems/assign cookies/description/?envType=problem-list-v2&envId=sorting

   Program:

```
int cmp(const void *a, const void *b)
{
    return *(int *)a - *(int *)b;
}

int findContentChildren(int* g, int gSize, int* s, int sSize)
{
    qsort(g, gSize, sizeof(int), cmp);
    qsort(s, sSize, sizeof(int), cmp);
    int result = 0;
    int i = 0;
    int j = 0;
while (i < gSize && j < sSize)
    {
        while (j < sSize && g[i] > s[j])
            ++j;
        if (i < gSize && j < sSize)
            ++result;
        ++i;
        ++j;
    }

    return result;
}
```

Skill Session:

1. Simple Sorting Given a list of numbers, you have to sort them in non decreasing order. Input Format The first line contains a single integer, N, denoting the number of integers in the list. The next N lines contain a single integer each, denoting the elements of the list. Output Format Output N lines, containing one integer each, in non-decreasing order. Link:https://www.codechef.com/practice/course/sorting/SORTING/proble ms/TSORT
Program:

```
#include <stdio.h>
int compare(const void *a,const void *b)
{
   return(*(int*)a-*(int*)b);
}
int main()
{
   int n;
   scanf("%d",&n);
   int a[n];
   for(int i=0;i<n;i++)
   {
      scanf("%d",&a[i]);
   }
   qsort(a,n,sizeof(int),compare);
   for(int i=0;i<n;i++)
   {
      printf("%d\n",a[i]);
   }
   return 0;
}
```

2. Squares of a sorted array Given an integer array nums sorted in non-decreasing order, return an array of the squares of each number sorted in non-decreasing order.
Link:
https://leetcode.com/problems/squares-of-a-sorted-array/
Program:

```
int* sortedSquares(int* nums, int numsSize, int* returnSize)
{
   int *res = malloc(sizeof(nums[0]) * numsSize);
   int start = 0;
   int end = numsSize - 1;
   for (int i = numsSize - 1; start <= end; i--) {
      int a = nums[start] * nums[start];
      int b = nums[end] * nums[end];
      if (a > b) {
         res[i] = a;
         start++;
      } else {
         res[i] = b;
```

```
            end--;
        }
    }
    *returnSize = numsSize;
    return res;
}
```

3. Score Sort(Shell Sort)
   You work in the examination department of a school, and you've been given a task to sort the test scores of students in ascending order. The scores are stored in a list where each element represents the score of a student. Input Format The first line contains the integer N, the size of the array. The next line contains N space-separated integers. Constraints • 1<=N<=1000 • -1000<=a[i]<=1000 Output Format Print the array as a row of space-separated integers in each iteration.
   Link:
   https://www.hackerrank.com/contests/ds-week 2/challenges/score-sortshell-sort
   Program:

```
#include<stdio.h>

void shellSort(int [],int);
void print(int [],int);

int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
        scanf("%d",&arr[i]);
    shellSort(arr,n);
    return 0;
}

void shellSort(int arr[], int n)
{
   for (int gap = n / 2; gap > 0; gap /= 2)
   {
    for (int j = gap; j < n; j += 1)
    {
      int i;
      int temp;
      for (i = j-gap;i>=0;i=i-gap)
      {
if(arr[i]>arr[i+gap])
            {
               temp=arr[i];
               arr[i]=arr[i+gap];
               arr[i+gap]=temp;
            }
            else
```

```
            {
               break;

            }
      }
        print(arr,n);
    }
  }
}

void print(int arr[],int n)
{
    for(int i=0;i<n;i++)
        printf("%d ",arr[i]);
    printf("\n");
}
```

4. Sort Even and Odd Indices Independently You are given a 0-indexed integer array nums. Rearrange the values of nums according to the following rules: • Sort the values at odd indices of nums in non-increasing order. For example, if nums = [4,1,2,3] before this step, it becomes 4,3,2,1] after. The values at odd indices 1 and 3 are sorted in non-increasing order. • Sort the values at even indices of nums in non-decreasing order. For example, if nums = [4,1,2,3] before this step, it becomes [2,1,4,3] after. The values at even indices 0 and 2 are sorted in non-decreasing order.
   Link:
   https://leetcode.com/problems/sort-even-and-odd-indices independently/description/
   Program:

```
int* sortEvenOdd(int* nums, int numsSize, int* returnSize) {
    int *arr=(int *)malloc(numsSize * sizeof(int));
     for(int i=0;i<numsSize;i++)
    {
        arr[i]=nums[i];
    }
    for(int i=0;i<numsSize;i+=2)
    {
        int key=arr[i];
        int j=i-2;
        while(j>=0 && arr[j]>key)
        {
            arr[j+2]=arr[j];
            j-=2;
        }
        arr[j+2]=key;
    }
    for(int i=1;i<numsSize;i+=2)
    {
        int key=arr[i];
```

```
        int j=i-2;
        while(j>=0 && arr[j]<key)
        {
            arr[j+2]=arr[j];
            j-=2;
        }
        arr[j+2]=key;
    }
    *returnSize=numsSize;
    return arr;
}
```

5. A post on facebook is said to be more popular if the number of likes on the post is strictly greater than the number of likes on some other post. In case the number of likes is same, the post having more comments is more popular. Given arrays A and B, each having size N, such that the number of likes and comments on the ith post are Ai and Bi respectively. Find out the most popular post. It is guaranteed that the number of comments on all the posts is distinct. Link: https://www.codechef.com/practice/course/sorting/SORTING/problems /FACEBOOK
   Program:

```
#include <stdio.h>

int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        int n;
        scanf("%d", &n);
        int a[n], b[n];
        for (int i = 0; i < n; i++) {
            scanf("%d", &a[i]);
        }
        for (int i = 0; i < n; i++) {
            scanf("%d", &b[i]);
        }
int maxLikes = a[0], maxComments = b[0], maxIndex = 0;
            for(int i=1;i<n;i++){
                if(a[i]>maxLikes || (a[i]==maxLikes && b[i]>maxComments)){
                    maxLikes = a[i];
                    maxComments = b[i];
                    maxIndex = i;
                }
            }
            printf("%d\n",maxIndex+1);
        }
```

```
        return 0;
}
```

6. Sort Integers by The Number of 1 Bits You are given an integer array arr. Sort the integers in the array in ascending order by the number of 1's in their binary representation and in case of two or more integers have the same number of 1's you have to sort them in ascending order. Return the array after sorting it.
   Link:
   https://leetcode.com/problems/sort-integers-by-the-number-of-1 bits/description/
   Program:

```
int count_ones(int a) {
    int count = 0;
    while(a > 0) {
        if((a & 0x01)) {
            count += 1;
        }

        a >>= 1;
    }

    return count;
}

int cmp(const void *a, const void* b) {
    int a_count_ones = count_ones(*(int*)a);
    int b_count_ones = count_ones(*(int*)b);
    if(a_count_ones == b_count_ones) {
        return (*(int*)a - *(int *)b);
    } else {
        return (a_count_ones - b_count_ones);
    }
}

int* sortByBits(int* arr, int arrSize, int* returnSize) {
    qsort(arr, arrSize, sizeof(int), cmp);
    *returnSize = arrSize;

    return arr;
}
```