

1.Imagine you have a tower of hanoi puzzle with four disks and each disk has a unique weight the disks are initially stacked in ascending order of weight .The objective is to move all the disks to first pin to last pin which is considered as destination Inorder to go with the problem we have measure constraints 1.Only one disk can be moved at a time. 2.A disk can only be placed on top of a larger disk or an empty pin. Write a function that calculates the minimum number of moves required to solve the Tower of Hanoi puzzle with four disks.

```
#include<stdio.h>
#include<math.h>
int Tow_of_Han(int n)
{
    if(n==0)
    {
        return 0;
    }
    return (int)(pow(2, n) - 1);
}
int main()
{
    int disks=4;
    int move=Tow_of_Han(disks);
    printf("Enter the minimum number of moves to solve tower of hanoi with %d disks\n",disks,move);
    return 0;
}
```

2 Sorting exam scores of a small class. A teacher wants to sort the exam scores of a small class of students in ascending order to identify the top performer.

```
#include <stdio.h>

void bubble_sort(int score[], int n) {
    for (int i = 0; i < n - 1; i++) {
        int swapped = 0;
        for (int j = 0; j < n - i - 1; j++) {
            if (score[j] > score[j + 1]) {
                int temp = score[j];
                score[j] = score[j + 1];
                score[j + 1] = temp;
                swapped = 1;
            }
        }
        if (swapped == 0) {
            break;
        }
    }
}
```

```

    }
}

int main() {
    int n;
    printf("Enter the number of exam scores: ");
    scanf("%d", &n);
    int ex_sco[n];
    printf("Enter the exam scores:\n");
    for (int i = 0; i < n; i++) {
        printf("Score of student %d: ", i + 1);
        scanf("%d", &ex_sco[i]);
    }
    printf("Scores before sorting:\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", ex_sco[i]);
    }
    printf("\n");
    bubble_sort(ex_sco, n);
    printf("Scores after sorting:\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", ex_sco[i]);
    }
    printf("\n");
    return 0;
}

```

3.Rotate array

```

def rotate(nums, k):
    k %= len(nums)
    if k == 0:
        return
    nums[:] = nums[-k:] + nums[:-k]

def main():
    nums = list(map(int, input("Enter elements of the array: ").split()))
    k = int(input("Enter number of rotations: "))

    rotate(nums, k)

    print("Rotated array:", ' '.join(map(str, nums)))

if __name__ == "__main__":
    main()

```

4.Merge two unsorted array

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

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

```
int main()
{
    int size1, size2;

    printf("Enter size of first array:\n ");
    scanf("%d", &size1);
    int arr1[size1];
    printf("Enter elements of first array:\n ");
    for (int i = 0; i < size1; i++) {
        scanf("%d", &arr1[i]);
    }

    printf("Enter size of second array:\n ");
    scanf("%d", &size2);
    int arr2[size2];
    printf("Enter elements of second array:\n ");
    for (int i = 0; i < size2; i++)
    {
        scanf("%d", &arr2[i]);
    }

    int merged[size1 + size2];
    for (int i = 0; i < size1; i++)
    {
        merged[i] = arr1[i];
    }
    for (int i = 0; i < size2; i++)
    {
```

```

        merged[size1 + i] = arr2[i];
    }

    bubbleSort(merged, size1 + size2);

    printf("Sorted merged array: ");
    for (int i = 0; i < size1 + size2; i++)
    {
        printf("%d ", merged[i]);
    }
    printf("\n");

    return 0;
}

```

5.Organizing the books on small shelf

```

#include <stdio.h>

void bubble_Sort(char titles[][100], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (strcmp(titles[j], titles[j + 1]) > 0)
            {
                char temp[100];
                for (int k = 0; k < 100; k++)
                {
                    temp[k] = titles[j][k];
                    titles[j][k] = titles[j + 1][k];
                    titles[j + 1][k] = temp[k];
                }
            }
        }
    }
}

int main()
{
    int n;

    printf("Enter number of books: ");
    scanf("%d", &n);
    char titles[n][100];
}

```

```
printf("Enter book titles:\n");
for (int i = 0; i < n; i++)
{
    scanf(" %s", titles[i]);
}

bubble_Sort(titles, n);

printf("Sorted book titles:\n");
for (int i = 0; i < n; i++)
{
    printf("%s\n", titles[i]);
}

return 0;
}
```