

## Question 1

### Source code:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n,i,*p,arr[n],j,count;
    printf("Enter number of elements:\n");
    scanf("%d",&n);
    p=(int*)malloc(n * sizeof(int));
    if(p == NULL)
    {
        printf("Memory is not allocated\n");
    }
    else
    {
        printf("Enter number elements of array you want to store:\n");
        for(i=0;i<n;++i)
        {
            scanf("%d",&*(p+i));
            arr[i]=-1;
        }
        printf("Elements of array are: \n");
        for(i=0;i<n;i++)
        {
            printf("%d\n",*(p+i));
        }
        for(i=0; i<n; i++)
        {
            count = 1;
            for(j=i+1; j<n; j++)
            {
                if(*(p+i)==*(p+j))

                {
                    count++;

                    arr[j] = 0;
                }
            }

            if(arr[i] != 0)
            {
                arr[i] = count;
            }
        }

        printf("\nFrequency of all elements of array : \n");
        for(i=0; i<n; i++)
        {
```

```
        if(arr[i] != 0)
        {
            printf("%d occurs %d times\n", *(p+i), arr[i]);
        }
    }
    free(p);
    return 0;
}
```

### ***Output:***

Enter number of elements:

4

Enter number elements of array you want to store:

1

3

4

4

Elements of array are:

1

3

4

4

Frequency of all elements of array :

1 occurs 1 times

3 occurs 1 times

4 occurs 2 times

## Question 2

### Source code:

```
#include<stdio.h>
#include<stdlib.h>
int main(int argc, char* argv[])
{
    int rows = 0;
    int cols = 0;
    int height = 0;
    int ***array;
    int r, c, h;
    printf ("3D Array has rows : ");
    scanf ("%d", &rows);
    printf ("3D Array has columns : ");
    scanf ("%d", &cols);
    printf ("3D Array has height : ");
    scanf ("%d", &height);
    array = (int ***) calloc (height,sizeof(int ***));
    for (h = 0; h < height; h++) {
        array[h] = (int **) calloc(rows,sizeof(int*));
        for (r = 0; r < rows; r++) {
            array[h][r] = (int *) calloc(cols,sizeof(int));
        }
    }
    for (h = 0; h < height; h++) {
        for (r = 0; r < rows; r++) {
            for (c = 0; c < cols; c++) {
                printf ("Enter Array Element [%d][%d][%d] : ", h, r, c);
                scanf ("%d", &array[h][r][c]);
            }
        }
    }
    printf("Printing 3D Array:\n");
    for (h = 0; h < height; h++) {
        printf("Height %d\n", h);
        for (r = 0; r < rows; r++) {
            for (c = 0; c < cols; c++) {
                printf("%.2d ", array[h][r][c]);
            }
            printf("\n");
        }
        printf("\n");
    }
    return 0;
}
```

### ***Output:***

```
3D Array has rows : 3
3D Array has columns : 2
3D Array has height : 1
Enter Array Element [0][0][0] : 1
Enter Array Element [0][0][1] : 2
Enter Array Element [0][1][0] : 1
Enter Array Element [0][1][1] : 4
Enter Array Element [0][2][0] : 6
Enter Array Element [0][2][1] : 7
Printing 3D Array:
Height 0
01 02
01 04
06 07
```

### Question 3

#### Source code:

```
#include <math.h>
#include <stdio.h>
float calculateSD(float data[]);
int product();
float mean();
float d[5];
int main() {
    int i;
    printf("Enter elements: ");
    for (i = 0; i < 5; ++i)
        scanf("%f", &d[i]);
    printf("\nStandard Deviation = %.2f", calculateSD(d));
    product();
    mean();
}

float calculateSD(float d[]) {
    float sum = 0.0, mean, SD = 0.0;
    int i;
    //int n;
    for (i = 0; i < 5; ++i) {
        sum += d[i];
    }
    mean = sum / 5;
    for (int i = 0; i < 5; ++i)
        SD += pow(d[i] - mean, 2);
    return sqrt(SD / 5);
}

float mean(){
    float mean;
    //int n;
    float sum = 0.0;
    for (int i = 0; i < 5; ++i) {
        sum += d[i];
    }
    mean = sum / 5;
    printf("\nmean = %f",mean );
}

int product(){
    int p = 1;
    //int n;
    int i;
    for (i=0;i<5;i++){
        p = p * d[i];
    }
    printf("\nproduct = %d ", p );
}
```

```
}
```

Output:

Enter elements: 4

6

2

4

9

Standard Deviation = 2.37

product = 1728

mean = 5.000000

#### Question 4

##### Source code:

```
#include <stdio.h>

void leftRotatebyOne(int arr[], int n);

void leftRotate(int arr[], int d, int n)
{
    int i;
    for (i = 0; i < d; i++)
        leftRotatebyOne(arr, n);
}

void leftRotatebyOne(int arr[], int n)
{
    int temp = arr[0], i;
    for (i = 0; i < n - 1; i++)
        arr[i] = arr[i + 1];
    arr[i] = temp;
}

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

int main()
{
    int arr[] = { 1, 2, 3, 4, 5, 6, 7 };
    leftRotate(arr, 2, 7);
    printArray(arr, 7);
    return 0;
}
```

##### Output:

3 4 5 6 7 1 2

## Question 5

### Source code:

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

int main()
{

    int *ptr, *ptr1;
    int n, i;


    printf("Enter number of Elements: \n");
    scanf("\n%d", &n);


    ptr = (int*)malloc(n * sizeof(int));

    ptr1 = (int*)calloc(n, sizeof(int));


    if (ptr == NULL || ptr1 == NULL) {
        printf("Memory not allocated.\n");
        exit(0);
    }
    else {

        printf("Memory successfully allocated using malloc.\n");


        printf("\nMemory successfully allocated using calloc.\n");

        free(ptr);
        printf("\nMalloc Memory successfully de-allocated.\n");

        free(ptr1);
        printf("\nCalloc Memory successfully de-allocated.\n");
    }

    return 0;
}
```



***Output:***

Enter number of Elements:

1 3 4 6 7

Memory successfully allocated using malloc.

Memory successfully allocated using calloc.

Malloc Memory successfully de-allocated.

Calloc Memory successfully de-allocated.