

Q1.

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

typedef struct
{
    char name[30];
    char gender[30];
    char designation[30];
    int salary;
    int gsalary;
} Employee;

int main()
{
    int i, n = 2, hra, da;

    Employee employees[n];

    //Taking each employee detail as input

    printf("Enter %d Employee Details \n \n", n);
    for (i = 0; i < n; i++)
    {

        printf("Employee %d:- \n", i + 1);
        //Name
        printf("Name: ");
        scanf("%s", employees[i].name);

        //Gender
        printf("Gender: ");
        scanf("%s", employees[i].gender);

        //Designation
        printf("Designation: ");
        scanf("%s", employees[i].designation);

        //Salary
        printf("Salary: ");
        scanf("%d", &employees[i].salary);
```

```

        da = 0.75 * employees[i].salary;
        hra = 0.25 * employees[i].salary;
        employees[i].gsalary = employees[i].salary + da + hra;

        printf("\n");
    }
    //Displaying Employee details

    printf("----- All Employees Details ----- \n");

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

        printf("Name \t: ");
        printf("%s \n", employees[i].name);

        printf("Gender \t: ");
        printf("%s \n", employees[i].gender);

        printf("Designation \t: ");
        printf("%s \n", employees[i].designation);

        printf("Gross Salary \t: ");
        printf("%d \n", employees[i].gsalary);

        printf("\n");
    }

    return 0;
}

```

Output

```

Enter 2 Employee Details

Employee 1:-
Name: Anupam
Gender: Male
Designation: SeniorDev
Salary: 20000

Employee 2:-
Name: Anuraag
Gender: Male
Designation: JuniorDev
Salary: 10000

----- All Employees Details -----
Name      : Anupam
Gender    : Male
Designation : SeniorDev
Gross Salary : 40000

Name      : Anuraag
Gender    : Male
Designation : JuniorDev
Gross Salary : 20000

```

Q2.

```
#include<stdio.h>

struct dist
{
    int km;
    int m;
};
void addition(struct dist,struct dist,struct dist *);

void main()
{
    struct dist d1,d2,d3;
    printf("Enter km and m for 1st system: ");
    scanf("%d %d",&d1.km,&d1.m);
    printf("\nEnter km and m for 2nd system: ");
    scanf("%d %d",&d2.km,&d2.m);
    addition(d1,d2,&d3);
    printf("\nAddition of two systems is %d km %d m",d3.km,d3.m);
}

void addition(struct dist d1,struct dist d2,struct dist *d3)
{
    (*d3).km=d1.km+d2.km;
    (*d3).m=d1.m+d2.m;
    if((*d3).m>=1000)
    {
        (*d3).km++;
        (*d3).m-=1000;
    }
}
```

Output

```
Enter km and m for 1st system: 1
20
```

```
Enter km and m for 2nd system: 4
60
```

```
Addition of two systems is 5 km 80 m
```

```
PS C:\Users\KIIIT\Documents\DSA LAB\DSA LAB2> |
```

Q3.

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

int a[5], pos, elem;
int n = 0;
void create();
void display();
void insert();
void del();
void sort();
void main()
{
    int choice;
    while (1)
    {
        printf("\n\n-----Select a Valid option max array size 5-----
");
        printf("\n=>1. Create an array of N integers");
        printf("\n=>2. Display of array elements");
        printf("\n=>3. Insert ELEM at a given POS");
        printf("\n=>4. Delete an element at a given POS");
        printf("\n=>5. Sort the Array");
        printf("\n=>6. Exit");
        printf("\nEnter your choice: ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                create();
                break;
            case 2:
                display();
                break;
            case 3:
                insert();
                break;
            case 4:
                del();
                break;
            case 5:
                sort();
                break;
            case 6:
                exit(0);
                break;
        }
    }
}
```

```

        sort();
        break;
    case 6:
        exit(1);
        break;
    default:
        printf("\nPlease enter a valid choice:");
    }
}
}
void create()
{
    int i;
    printf("\nEnter the number of elements: ");
    scanf("%d", &n);
    printf("\nEnter the elements: ");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
}
void display()
{
    int i;
    if (n == 0)
    {
        printf("\nNo elements to display");
        return;
    }
    printf("\nArray elements are: ");
    for (i = 0; i < n; i++)
        printf("%d\t", a[i]);
}
void insert()
{
    int i;
    if (n == 5)
    {
        printf("\nArray is full. Insertion is not possible");
        return;
    }
    do
    {

```

```

        printf("\nEnter a valid position where element to be inserted:
");
        scanf("%d", &pos);
    } while (pos > n);
    printf("\nEnter the value to be inserted:  ");
    scanf("%d", &elem);
    for (i = n - 1; i >= pos; i--)
    {
        a[i + 1] = a[i];
    }
    a[pos] = elem;
    n = n + 1;
    display();
}
void del()
{
    int i;
    if (n == 0)
    {
        printf("\nArray is empty and no elements to delete");
        return;
    }
    do
    {
        printf("\nEnter a valid position from where element to be deleted:
");
        scanf("%d", &pos);
    } while (pos >= n);
    elem = a[pos];
    printf("\nDeleted element is : %d \n", elem);
    for (i = pos; i < n - 1; i++)
    {
        a[i] = a[i + 1];
    }
    n = n - 1;
    display();
}
void sort()
{
    int temp = 0;

    //Calculate length of array a
    int length = sizeof(a) / sizeof(a[0]);

```

```

//Displaying elements of original array
printf("Elements of original array: \n");
for (int i = 0; i < length; i++)
{
    printf("%d ", a[i]);
}
//Sort the array in ascending order
for (int i = 0; i < length; i++)
{
    for (int j = i + 1; j < length; j++)
    {
        if (a[i] > a[j])
        {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
}

printf("\n");

printf("Elements of array sorted in ascending order: \n");
for (int i = 0; i < length; i++)
{
    printf("%d ", a[i]);
}
}

```

Output

```

-----Select a Valid option max array size 5-----
=>1. Create an array of N integers
=>2. Display of array elements
=>3. Insert ELEM at a given POS
=>4. Delete an element at a given POS
=>5. Sort the Array
=>6. Exit
Enter your choice: 1

Enter the number of elements: 3

Enter the elements: 1

```

2
0

-----Select a Valid option max array size 5-----

- =>1. Create an array of N integers
- =>2. Display of array elements
- =>3. Insert ELEM at a given POS
- =>4. Delete an element at a given POS
- =>5. Sort the Array
- =>6. Exit

Enter your choice: 2

Array elements are: 1 2 0

-----Select a Valid option max array size 5-----

- =>1. Create an array of N integers
- =>2. Display of array elements
- =>3. Insert ELEM at a given POS
- =>4. Delete an element at a given POS
- =>5. Sort the Array
- =>6. Exit

Enter your choice: 3

Enter a valid position where element to be inserted: 0

Enter the value to be inserted: 6

Array elements are: 6 1 2 0

-----Select a Valid option max array size 5-----

- =>1. Create an array of N integers
- =>2. Display of array elements
- =>3. Insert ELEM at a given POS
- =>4. Delete an element at a given POS
- =>5. Sort the Array
- =>6. Exit

Enter your choice: 4

Enter a valid position from where element to be deleted: 3

Deleted element is : 0


```
Array elements are: 6    1    2

-----Select a Valid option max array size 5-----
=>1. Create an array of N integers
=>2. Display of array elements
=>3. Insert ELEM at a given POS
=>4. Delete an element at a given POS
=>5. Sort the Array
=>6. Exit
Enter your choice: 5
Elements of original array:
6 1 2 0 0
Elements of array sorted in ascending order:
0 0 1 2 6

-----Select a Valid option max array size 5-----
=>1. Create an array of N integers
=>2. Display of array elements
=>3. Insert ELEM at a given POS
=>4. Delete an element at a given POS
=>5. Sort the Array
=>6. Exit
Enter your choice: 2

Array elements are: 0    0    1

-----Select a Valid option max array size 5-----
=>1. Create an array of N integers
=>2. Display of array elements
=>3. Insert ELEM at a given POS
=>4. Delete an element at a given POS
=>5. Sort the Array
=>6. Exit
Enter your choice: 6
```