

# PF LAB 07

## TASK 02:

Note: (C File is not available because I forgot to copy them at the library and they have been deleted now.)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int i,n, d, input[7];
6
7      printf("Enter the value of d: ");
8      scanf("%d", &d);
9
10
11     printf("Enter the length of array: ");
12     scanf("%d", &n);
13     for (i=0; i<n; i++) {
14         printf("Enter the element %d: ", i+1);
15         scanf("%d", &input[i]);
16     }
17
18     for (i=d; i<7; i++) {
19         printf("%d, ", input[i]);
20     }
21
22     for (i=0; i<d;i++) {
23         printf("%d, ", input[i]);
24     }
25
26     |
27     return 0;
28 }
```

```
Enter the value of d: 2
Enter the length of array: 7
Enter the element 1: 1
Enter the element 2: 2
Enter the element 3: 3
Enter the element 4: 4
Enter the element 5: 5
Enter the element 6: 6
Enter the element 7: 7
3, 4, 5, 6, 7, , 1, 2

...Program finished with exit code 0
Press ENTER to exit console.□
```

## TASK 05:

Note: (C File is not available because I forgot to copy them at the library and they have been deleted now.)

```
1  #include <stdio.h>
2  int main () {
3
4      int i,j, r=9;
5      int px = r / 2 +1;
6
7      for (i=1; i<=r;i++) {
8          for (j=1;j<=r;j++) {
9              if (j==px || j==r-px+1)
10                 printf("*");
11             else
12                 printf(" ");
13         }
14
15         if (i<= r/2) {
16             px--;
17         }
18         else {
19             px++;
20         }
21         printf("\n");
22     }
23 }
24
25 return 0;
26 }
```



input

```
*
* *
*   *
*     *
*       *
*         *
*           *
*             *
*               *
```

## TASK 06:

Q6.c

```
1 #include <stdio.h>
2
3 int main () {
4
5     int freq[100]={0,0,0,0,0}, n, i, j, k=i-1, count, arr[100];
6
7     printf("enter the length of the array: ");
8     scanf("%d", &n);
9
10
11     for (i=0; i<n; i++) {
12         printf("enter the element %d of the array: ", i+1);
13         scanf("%d", &arr[i]);
14     }
15
16
17     for (i=0; i<n; i++) { // {2,4,2,3,5,5,4,4}
18         count=1;
19         for (j=i+1; j<n; j++) {
20             if (arr[i]==arr[j]) {
21                 count++;
22                 freq[j] = -1; // ye element ko processed mark krdega
23             }
24         }
25         if (freq[i] != -1) { // ye check krega if the number is repeated or not
26             printf("Frequency of %d is %d\n", arr[i], count);
27         }
28     }
29
30
31
32     return 0;
33 }
```

```
C:\Users\3TEE\Desktop\Q6.exe
enter the length of the array: 8
enter the element 1 of the array: 2
enter the element 2 of the array: 4
enter the element 3 of the array: 2
enter the element 4 of the array: 3
enter the element 5 of the array: 5
enter the element 6 of the array: 5
enter the element 7 of the array: 4
enter the element 8 of the array: 4
Frequency of 2 is 2
Frequency of 4 is 3
Frequency of 3 is 1
Frequency of 5 is 2

-----
Process exited after 12.61 seconds with return value 0
Press any key to continue . . .
```

## TASK 07:

```
Q7.c
1  include <stdio.h>
2
3  int main () {
4
5      int i, j, k, null[2][2] = {{0,0},{0,0}}, matrix1[2][2], matrix2[2][2], multiply[2][2]={0,0,0,0}, result[2][2]={0,0,0,0}
6
7      printf("enter the first matrix: ");
8      for (i=0;i<2;i++) {
9          for (j=0;j<2;j++) {
10             scanf("%d", &matrix1[i][j]);
11         }
12     }
13
14     printf("enter the second matrix: ");
15     for (i=0;i<2;i++) {
16         for (j=0;j<2;j++) {
17             scanf("%d", &matrix2[i][j]);
18         }
19     }
20
21     // |a1  b1|  |a2  b2|  -> |a1*a2 + b1*c2  a1*b2 + b1*d2|
22     // |c1  d1|  |c2  d2|  -> |c1*a2 + d1*c2  c1*b2 + d1*d2|
23
24     for (i=0;i<2;i++) {
25         for (j=0;j<2;j++) {
26             result[i][j]=0;
27             for (k=0;k<2;k++) {
28                 result[i][j] += matrix1[i][k] * matrix2[k][j];
29             }
30             // result[i][j] = matrix1[i][j] * matrix2[j][i] + matrix1[i][k] * matrix2[k][i];
31             // printf("%d ", result[i][j]);
32         }
33     }
34
35     // printf("%d ", result[i][j]);
36     // printf("\n");
37     // result[i][j] = result [i][j] + multiply [i][j];
38
39     }
40
41     }
42
43     }
44
45     }
46
47     return 0;
48
```

```
31 // printf("%d ", result[i][j]);
32 // printf("\n");
33 // result[i][j] = result [i][j] + multiply [i][j];
34 }
35 }
36
37 }
38
39 printf("Resultant matrix = \n");
40 for (i=0;i<2;i++) {
41     for (j=0;j<2;j++) {
42         printf("%d ", result[i][j]);
43     }
44     printf("\n");
45 }
46
47 return 0;
48
```

```
C:\Users\3TEE\Desktop\Q7.exe
enter the first matrix: 1
2
3
4
enter the second matrix: 1
2
3
4
Resultant matrix =
7 10
15 22

-----
Process exited after 7.029 seconds with return value 0
Press any key to continue . . .
```

## TASK 10:

```
Q7.c Q10.c [Untitled3]
4 int i, j, N, S, arr[N], right=0, left=0, sum=0;
5 printf("Enter the size of the array: ");
6 scanf("%d", &N);
7
8
9 printf("Enter the elements of the array: ");
10 for (i=0; i<N; i++) {
11     scanf("%d", &arr[i]);
12 }
13
14 printf("Enter the target sum (S): ");
15 scanf("%d", &S);
16
17
18
19 while (right < N) {
20     sum += arr[right];
21
22     while (sum > S) {
23         sum -= arr[left];
24         left++;
25     }
26
27     if (sum == S) {
28         printf("Subarray found between indices %d and %d\n", left, right);
29         printf("Subarray elements: ");
30         for (i = left; i <= right; i++) {
31             printf("%d ", arr[i]);
32         }
33         return 0;
34     }
35
36     right++;
37 }
38
39 printf("No subarray found with sum equal to %d\n", S);
40 return 0;
```

```
C:\Users\3TEE\Desktop\Q10.exe
Enter the size of the array: 5
Enter the elements of the array: 1
2
3
4
5
Enter the target sum (S): 12
Subarray found between indices 1 and 3
Subarray elements: 2 3 7
-----
Process exited after 18.43 seconds with return value 0
Press any key to continue . . .
```

## TASK 09:

```
Q7.c Q10.c q9.c
1 #include <stdio.h>
2
3 int main () {
4     /*I am assuming 3 branches and 3 mobile phones, that's why [3][3]*/
5     int i, j, branches, mobiles, bill[3][3], total_bill, max_total_bill=0, max_bill_branchID, max_bill_mobileID, max_phone_bill;
6
7     printf("enter the total number of branches: ");
8     scanf("%d", &branches);
9     printf("enter the total number of mobile phones: ");
10    scanf("%d", &mobiles);
11    for (i=0; i<branches; i++) {
12        for (j=0; j<mobiles; j++) {
13            printf("enter the phone bill for branch ID %d and mobile ID %d: ", i+1, j+1);
14            scanf("%d", &bill[i][j]);
15        }
16    }
17
18    // Total bill for each branches
19    printf("\n");
20    for (i=0; i<branches; i++) {
21        total_bill = 0;
22        for (j=0; j<mobiles; j++) {
23            total_bill += bill[i][j];
24        }
25        printf("Total bill for branch %d: %d\n", i+1, total_bill);
26    }
27
28    // Total bill for all branches
29    total_bill = 0;
30    for (i=0; i<branches; i++) {
31        for (j=0; j<mobiles; j++) {
32            total_bill += bill[i][j];
33        }
34    }
35    printf("Total bill for all branches: %d", total_bill);
36
37
38    // Branch ID where max bill
39    printf("\n");
40    for (i=0; i<branches; i++) {
41        total_bill = 0;
42        for (j=0; j<mobiles; j++) {
43            total_bill += bill[i][j];
44            // if (max_bill < total_bill) {
45            //     max_bill = total_bill;
46            //     max_bill_branchID = i + 1;
47            // }
48        }
49        if (max_total_bill < total_bill) {
50            max_total_bill = total_bill;
51            max_bill_branchID = i + 1;
52        }
53    }
54    printf("Maximum total bill reported at branch ID %d: %d\n", max_bill_branchID, max_total_bill);
55
56    // Branch and Mobile Phone IDs where bill is highest of all mobile phones.
57    max_phone_bill = 0;
58    for (i=0; i<branches; i++) {
59        total_bill = 0;
60        for (j=0; j<mobiles; j++) {
61            if (max_phone_bill < bill[i][j]) {
62                max_phone_bill = bill[i][j];
63                max_bill_branchID = i + 1;
64                max_bill_mobileID = j + 1;
65            }
66        }
67    }
68    printf("Maximum phone bill reported at branch ID %d & mobile phone ID %d: %d", max_bill_branchID, max_bill_mobileID, max_phone_bill);
69
70    return 0;
71 }
72 }
```

C:\Users\3TEE\Desktop\q9.exe

```
enter the total number of branches: 3
enter the total number of mobile phones: 3
enter the phone bill for branch ID 1 and mobile ID 1: 51000
enter the phone bill for branch ID 1 and mobile ID 2: 56000
enter the phone bill for branch ID 1 and mobile ID 3: 50000
enter the phone bill for branch ID 2 and mobile ID 1: 3500
enter the phone bill for branch ID 2 and mobile ID 2: 45000
enter the phone bill for branch ID 2 and mobile ID 3: 36600
enter the phone bill for branch ID 3 and mobile ID 1: 25000
enter the phone bill for branch ID 3 and mobile ID 2: 25550
enter the phone bill for branch ID 3 and mobile ID 3: 45000

Total bill for branch 1: 157000
Total bill for branch 2: 85100
Total bill for branch 3: 95550
Total bill for all branches: 337650
Maximum total bill reported at branch ID 1: 157000
Maximum phone bill reported at branch ID 1 & mobile phone ID 2: 56000
-----
Process exited after 30.05 seconds with return value 0
Press any key to continue . . .
```

