Birla Institute of Technology & Science, Pilani, Hyderabad Campus

First Semester 2020-2021
Computer Programming [CS F111] Lab 8

Practice Programs:

1. Write a C-program to input elements in an array and find frequency of each element in array.
Hint: read the size of the array before hand.

Sample Input/Output:

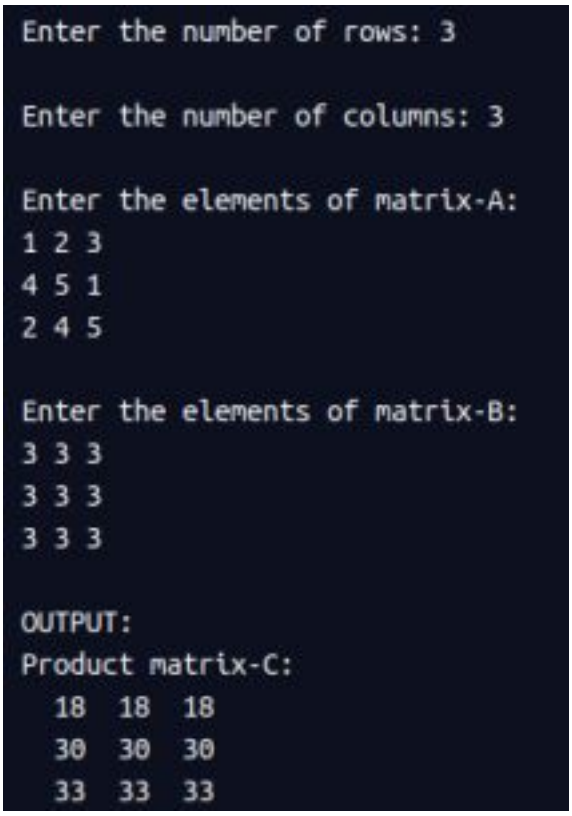
```
Enter the size of the array: 5

Enter the element-1: 1

Enter the element-2: 3

Enter the element-3: 2

Enter the element-4: 2

Enter the element-5: 1


OUTPUT:
1 Occurs 2 Times
3 Occurs 1 Times
2 Occurs 2 Times
```

Code:

```c
#include <stdio.h>
int main()
{
    int arr[10], FreqArr[10], i, j, Count, Size;
    printf ("\nEnter the size of the array: ");
    scanf("%d", &Size); //input size of the array
    for (i = 0; i < Size; i++)
    {
        printf("\nEnter the element-%d: ", i+1);
        scanf("%d", &arr[i]); //input the elements
        FreqArr[i] = -1;
    }
    for (i = 0; i < Size; i++)
    {
        Count = 1;
        for(j = i + 1; j < Size; j++)
        {
            if(arr[i] == arr[j])
            {
                Count++; //calculating the frequency of
                    same elements
                FreqArr[j] = 0;
            }
        }
        if(FreqArr[i] != 0)
        {
            FreqArr[i] = Count;
        }
    }
    printf("\n\nOUTPUT:\n");
    for (i = 0; i < Size; i++)
    {
        if(FreqArr[i] != 0)
        {
            printf("%d Occurs %d Times \n", arr[i],
                FreqArr[i]);
        }
    }
    return 0;
}
```

2. Write a C-program that will read the elements of matrices A and B (consider both of them to be square and of same size), and then produces a product matrix C. Hint: Take first and second input as number of rows and columns for both A and B. After that input matrix elements for A and B.

Sample Input/Output:



Code:

```c
#include<stdio.h>
#define MAX 10
int main()
{
    int a[MAX][MAX],b[MAX][MAX],c[MAX][MAX];
    int i,j,k,row,col;

    printf("\nEnter the number of rows: ");
    scanf("%d",&row); //number of rows
    printf("\nEnter the number of columns: ");
    scanf("%d",&col); //number of columns

    printf("\nEnter the elements of matrix-A: \n");
    for(i=0;i<row;i++)
        for(j=0;j<col;j++)
            scanf("%d",&a[i][j]);//matrix elements of A

    printf("\nEnter the elements of matrix-B: \n");
    for(i=0;i<row;i++)
        for(j=0;j<col;j++)
            scanf("%d",&b[i][j]); //matrix elements of B

    for(i=0;i<row;i++)
        for(j=0;j<col;j++)
        {
            c[i][j]=0;
            for(k=0;k<col;k++) //product matrix C
                c[i][j]=c[i][j]+(a[i][k]*b[k][j]);
        }

    printf("\nOUTPUT:\nProduct matrix-C: \n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
            printf("%4d",c[i][j]); //print matrix C
        printf("\n");
    }
    return 0;
}
```

3. An election is contested by 5 candidates. The candidate is numbered are 1 to 5 and the voting is done by marking the candidate number on the ballot paper.

Write a program to read the ballots and count the votes casted for each candidate using an array variable count.
In case, a number, read is outside the range 1 to 5, the ballot should be considered as a 'spoilt ballot' and the program should also count the number of spoilt ballots. Hint: first read the number of voters who casted their votes.

Sample Input/Output:

```
Enter the number of voters: 6

Vote-1: 1

Vote-2: 2

Vote-3: 9

Vote-4: 2

Vote-5: 3

Vote-6: 4

Candidate-1 Votes: 1
Candidate-2 Votes: 2
Candidate-3 Votes: 1
Candidate-4 Votes: 1
Candidate-5 Votes: 0
The number of spoilt votes: 1
```

Code:

```c
#include<stdio.h>
int main()
{
    int count[5], candidate, spoilt=0, voters;
    printf("\nEnter the number of voters: ");
    scanf("%d", &voters);

    for(int i=0;i<5;i++) //initialization
        count[i]=0;

    for(int i=0;i<voters;i++)
    {
        printf("\nVote-%d: ", i+1);
        scanf("%d",&candidate);
        switch(candidate)
        {
            case 1: count[0]=count[0]+1;
                    break;
            case 2: count[1]=count[1]+1;
                    break;
            case 3: count[2]=count[2]+1;
                    break;
            case 4: count[3]=count[3]+1;
                    break;
            case 5: count[4]=count[4]+1;
                    break;
            default: ++spoilt;
                    break;
        }
    }

    for(int i=0;i<5;i++)
        printf("\nCandidate-%d\tVotes: %d",i+1, count[i]);

    printf("\nThe number of spoilt votes: %d\n", spoilt);
    return 0;
}
```

Exercise Problems for Submission:

1. Modify the practice program-1 to print the elements in ascending order. Your program should print the number along with its frequency.

2. Modify the practice program-2 to produce the product of two matrices that may have different sizes, but confirmable for multiplication. For example, A is of size 2X3 and B is of size 3X2.

3. Modify the practice program-3 to print the winner among 1, 2, 3, 4, and 5. In case, multiple candidates obtained the highest votes, then print all their names/IDs. In case, the number of spoilt votes is the highest, then print **TRUMP WON**.

************ALL THE BEST************

NOTE: Upload the screenshots of the Exercise programs along with the displayed results into your corresponding Google Classroom.

PATH to Submit the Screenshots:
Google Classroom --> Classwork --> View Assignment --> Create/Upload