Q1

```c
// WAP to add and subtract two 3 X 3 matrices.

#include <stdio.h>

int main(){
    int arr1[3][3]={1,2,3,4,5,6,7,8,9};
    int arr2[3][3]={9,8,7,6,5,4,3,2,1};

    printf("Sum of two matrices :\n");
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            printf("%d ", arr1[i][j]+arr2[i][j]);
        }
        printf("\n");
    }

    printf("\nSubstraction of two matrices :\n");
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            printf("%d ", arr1[i][j]-arr2[i][j]);
        }
        printf("\n");
    }

    return 0;
}

/*
Sum of two matrices :
10 10 10
10 10 10
10 10 10

Substraction of two matrices :
-8 -6 -4
-2 0 2
4 6 8
*/
```

Q2

```c
// WAP to multiply two 3 χ 3 matrices.

/*
|1   2   3|       |9   8   7|
|4   5   6|   χ   |6   5   4|
|7   8   9|       |3   2   1|
*/

#include <stdio.h>

int main(){
    int arr1[3][3]={1,2,3,4,5,6,7,8,9};
    int arr2[3][3]={9,8,7,6,5,4,3,2,1};
    int arr12[3][3];

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {   arr12[i][j]=0;
            for (int k = 0; k < 3; k++)
            {
               arr12[i][j] += arr1[i][k] * arr2[k][j];
            //    printf("i=%d \t j=%d \t k=%d \n",i,j,k);
            }

        }
    }
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            printf("%d\t",arr12[i][j]);
        }
        printf("\n");
    }

    return 0;
}

/*
30      24      18
84      69      54
138     114     90
*/
```

```c
// WAP to input a 4 X 4 matrix and print the diagonal elements.

#include <stdio.h>

int main(){
    int arr[4][4], sum=0;
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            printf("Enter the value for index (%d, %d) : ", i, j);
            scanf("%d", &arr[i][j]);
        }
    }

    for (int i = 0; i < 4; i++)
    {
        for (int  j = 0; j < 4; j++)
        {
            if(i==j){
                printf("%d", arr[i][j]);
            }
            else{
                printf("\t");
            }
        }
        printf("\n");
    }

    return 0;
}

/*
Enter the value for index (0, 0) : 1
Enter the value for index (0, 1) : 2
Enter the value for index (0, 2) : 3
Enter the value for index (0, 3) : 4
Enter the value for index (1, 0) : 5
Enter the value for index (1, 1) : 6
Enter the value for index (1, 2) : 7
Enter the value for index (1, 3) : 8
Enter the value for index (2, 0) : 9
Enter the value for index (2, 1) : 10
Enter the value for index (2, 2) : 11
Enter the value for index (2, 3) : 12
Enter the value for index (3, 0) : 13
```

Q3

```
Enter the value for index (3, 1) : 14
Enter the value for index (3, 2) : 15
Enter the value for index (3, 3) : 16
1
        6
                11
                        16
*/
```

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

int main()
{

    int top = -1;
    int size = 5;
    int i;

    int arr[size], choice, data;
    while (top+1 <= size)
    {

        printf("\n------\n1. Push\n2. Pop\n3. Traverse\n\nEnter choice
: ");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            if(size - 1 == top){
                printf("Can't push value. Stack is full.");
                exit(0);
            }
            else{
                printf("Enter value : ");
                scanf("%d", &data);
                arr[top+1]=data;
                top++;
            }
            break;

        case 2:
            if(top == -1){
                printf("Stack is empty. Value can't be popped.");
                exit(0);
            }
            else{
                printf("\nElement removed : %d\n", arr[top]);
                top--;
            }
            break;

        case 3:
            if(top>=0){
```

Q4

```c
                i=0;
                do
                {
                    printf("%d ", arr[i]);
                    i++;
                } while (i<=top);
            }
            else{
                printf("No value to traverse.");
                exit(0);
            }
            break;
        };
    }

    return 0;
}

/*
------
1. Push
2. Pop
3. Traverse

Enter choice : 1
Enter value : 25

------
1. Push
2. Pop
3. Traverse

Enter choice : 1
Enter value : 30

------
1. Push
2. Pop
3. Traverse

Enter choice : 1
Enter value : 35

------
1. Push
2. Pop
```

```
3. Traverse

Enter choice : 2

Element removed : 35

-------
1. Push
2. Pop
3. Traverse

Enter choice : 3
25 30
*/
```