

```
#include<stdio.h>
int i,j,k,n;
int a[10][10];
int b[10][10];
int sum[10][10];
int sub[10][10];
int mul[10][10];
int tr[10][10];
int p_sum=0;
int nonP_sum=0;
int flag=1;
int row=0;
int column=0;
int add(int a[10][10],int b[10][10]){
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            sum[i][j]=a[i][j]+b[i][j];
        }
    }
    printf("the sum matrix is:\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("%d\t",sum[i][j]);
        }
        printf("\n");
    }
    printf("\n");
}
int subtract(int a[10][10],int b[10][10])
{
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            sub[i][j]=a[i][j]-b[i][j];
        }
    }
    printf("the resultant matrix after subtraction is:\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
```

```
        printf("%d\t",sub[i][j]);
    }
    printf("\n");
}
printf("\n");
}
int multiply(int a[10][10],int b[10][10])
{
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            for(k=0;k<n;k++)
            {
                mul[i][j]+= a[i][k]*b[k][j];
            }
        }
    }
    printf("the resultant matrix after multiplication is:\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("%d\t",mul[i][j]);
        }
        printf("\n");
    }
    printf("\n");
}
int add_principal(int a[10][10])
{
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            while(j<n)
            {
                if(i==j)
                {
                    p_sum+=a[i][j];
                }
                else{
                    nonP_sum+=a[i][j];
                }
            }
        }
    }
}
```

```
        j++;
    }

    break;
}

printf("the sum of principal diagonal elements:\n");
printf("%d\n",p_sum);
printf("the sum of non principal diagonal elements:\n");
printf("%d\n",nonP_sum);
printf("\n");
}

int add_row_column(int a[10][10])
{
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            row += a[i][j];
        }
    }
    for(j=0;j<n;j++)
    {
        for(i=0;i<n;i++)
        {
            column += a[i][j];
        }
    }
    printf("The sum of rows is %d\n",row);
    printf("the sum of column elements is %d",column);
    printf("\n");
}

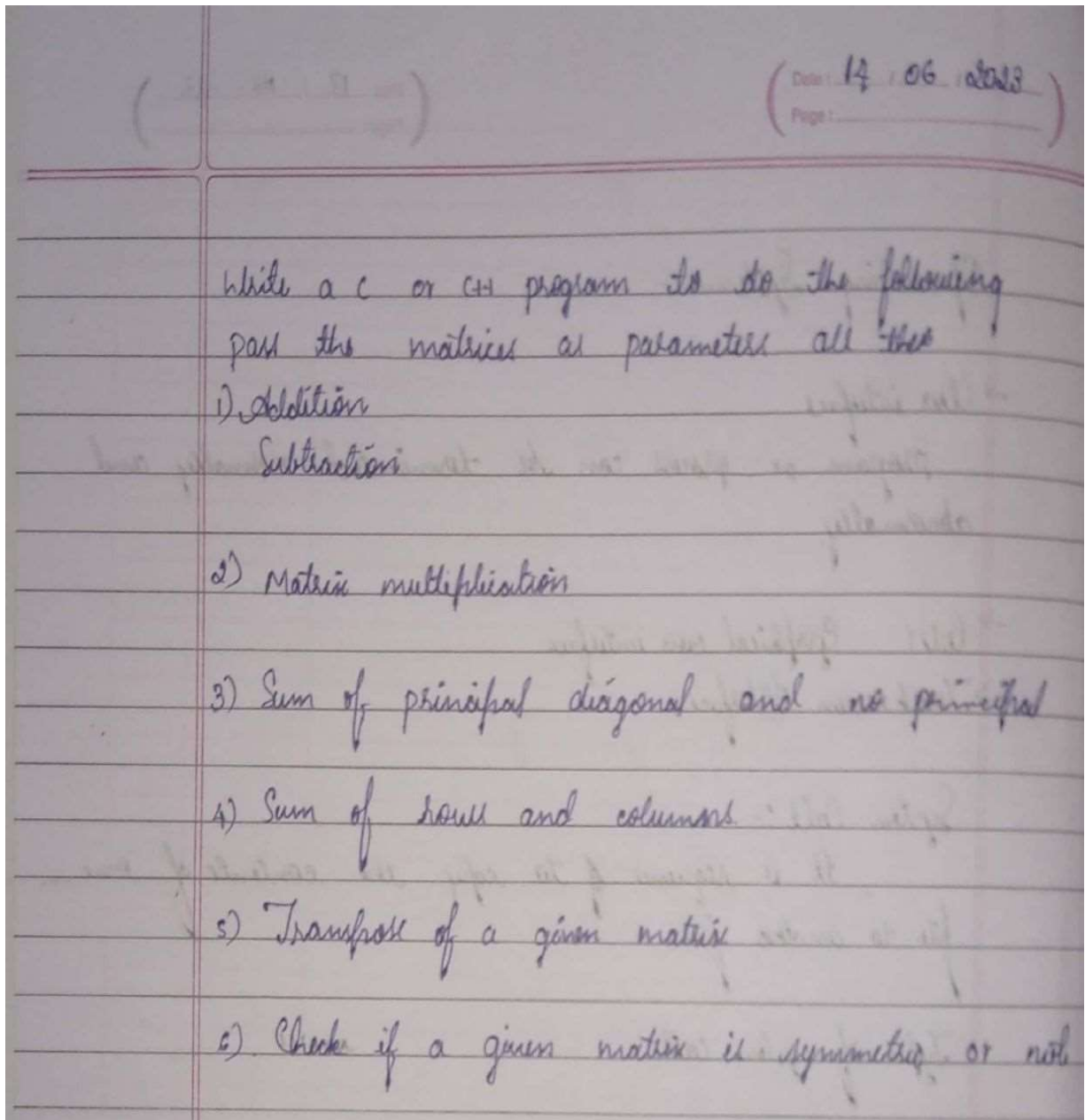
int transpose(int a[10][10])
{
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            tr[i][j]=a[j][i];
        }
    }
    printf("the transpose of A matrix:\n");
    for(i=0;i<n;i++)
    {
```

```
        for(j=0;j<n;j++)
        {
            printf("%d\t",tr[i][j]);
        }
        printf("\n");
    }
    printf("\n");
}
int symmetric(int a[10][10])
{
    transpose(a);
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++) {
            if(a[i][j]!=tr[i][j]) {
                flag = 0;
                break;
            }
        }
    }
    if(flag==1) {
        printf("Given matrix is symetric\n");
    }
    else{
        printf("Given matrix is not symmetric\n");
    }
    printf("\n");
}
void main()
{
    printf("Enter the size of n\n");
    scanf("%d",&n);
    printf("Enter the elements of A matrix\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Enter the elements of B matrix\n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            scanf("%d",&b[i][j]);
```

```

    }
}
printf("A matrix is\n");
for(i=0;i<n;i++){
    for(j=0;j<n;j++)
    {
        printf("%d\t",a[i][j]);
    }
    printf("\n");
}
printf("B matrix is\n");
for(i=0;i<n;i++){
    for(j=0;j<n;j++)
    {
        printf("%d\t",b[i][j]);
    }
    printf("\n");
}
int c;
while(1){
    printf("1.Addition\n2.Subtraction\n3.multiplication\n4.sum of principal and non principal
diagonal\n5. sum of rows and columns\n 6.transpose\n7.Symmetric\n");
    printf("Enter your choice:\n");
    scanf("%d",&c);
    switch(c){
        case 1:add(a,b);
        break;
        case 2:subtract(a,b);
        break;
        case 3:multiply(a,b);
        break;
        case 4:add_principal(a);
        break;
        case 5:add_row_column(a);
        break;
        case 6:transpose(a);
        break;
        case 7:symmetric(a);
        break;
        default:printf("Incorrect choice");
        exit(0);
    }
}
}
}

```

**Observation:**

( Date : \_\_\_\_ / \_\_\_\_ / \_\_\_\_ )  
Page : \_\_\_\_ )

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

1+6

1	2	1	2
3	4	3	4

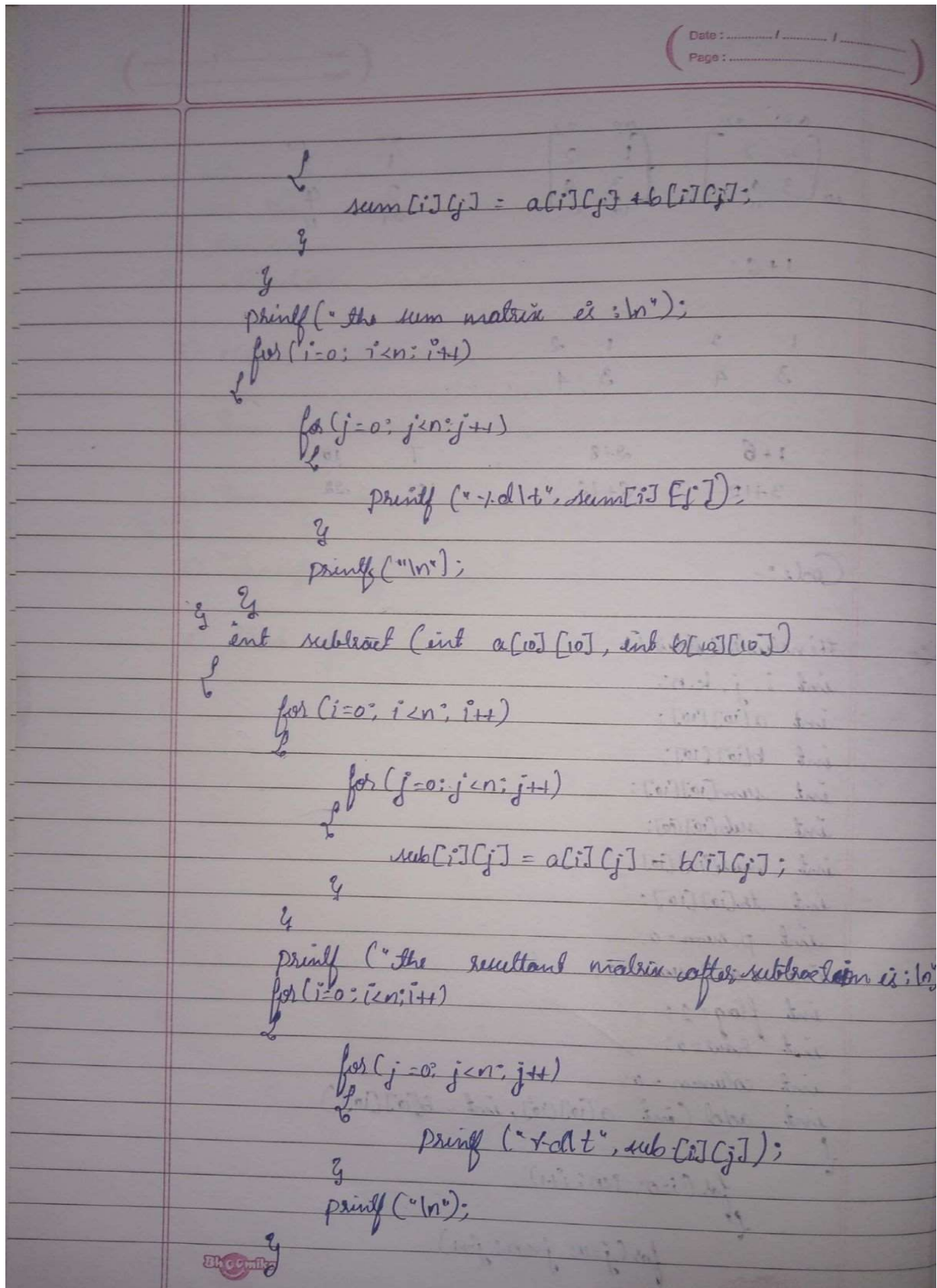
$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 6 & 11 \end{bmatrix}$$

$$\begin{matrix} 1+6 & 2+8 \\ 3+12 & 6+16 \end{matrix}$$

$$\begin{matrix} 7 & 10 \\ 15 & 22 \end{matrix}$$

Code:-

```
#include <stdio.h>
int i, j, k, n;
int a[10][10];
int b[10][10];
int sum[10][10];
int sub[10][10];
int mult[10][10];
int tr[10][10];
int psum = 0;
int nonPsum = 0;
int flag = 1;
int row = 0;
int column = 0;
int add(int a[10][10], int b[10][10])
{
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
```





( Date : ..... / ..... / ..... )  
Page : .....

```

}
int multiply(int a[10][10], int b[10][10])
{
    for(i=0; i<n; i++)
    {
        for(j=0; j<n; j++)
        {
            for(k=0; k<n; k++)
            {
                mul[i][j] += a[i][k] * b[k][j];
            }
        }
    }

    printf("The resultant matrix after multiplication is: \n");
    for(i=0; i<n; i++)
    {
        for(j=0; j<n; j++)
        {
            printf("%d\t", mul[i][j]);
        }
        printf("\n");
    }
}

int add_principal(int a[10][10])
{
    for(i=0; i<n; i++)
    {
        for(j=0; j<n; j++)
        {
            while(j<n)
            {
                if(i==j)
                {
                    p_sum += a[i][j];
                }
                else
                {
                    nonp_sum += a[i][j];
                }
                j++;
            }
        }
    }
}

```

Bhoomika

( Date: \_\_\_\_\_ )  
( Page: \_\_\_\_\_ )

```

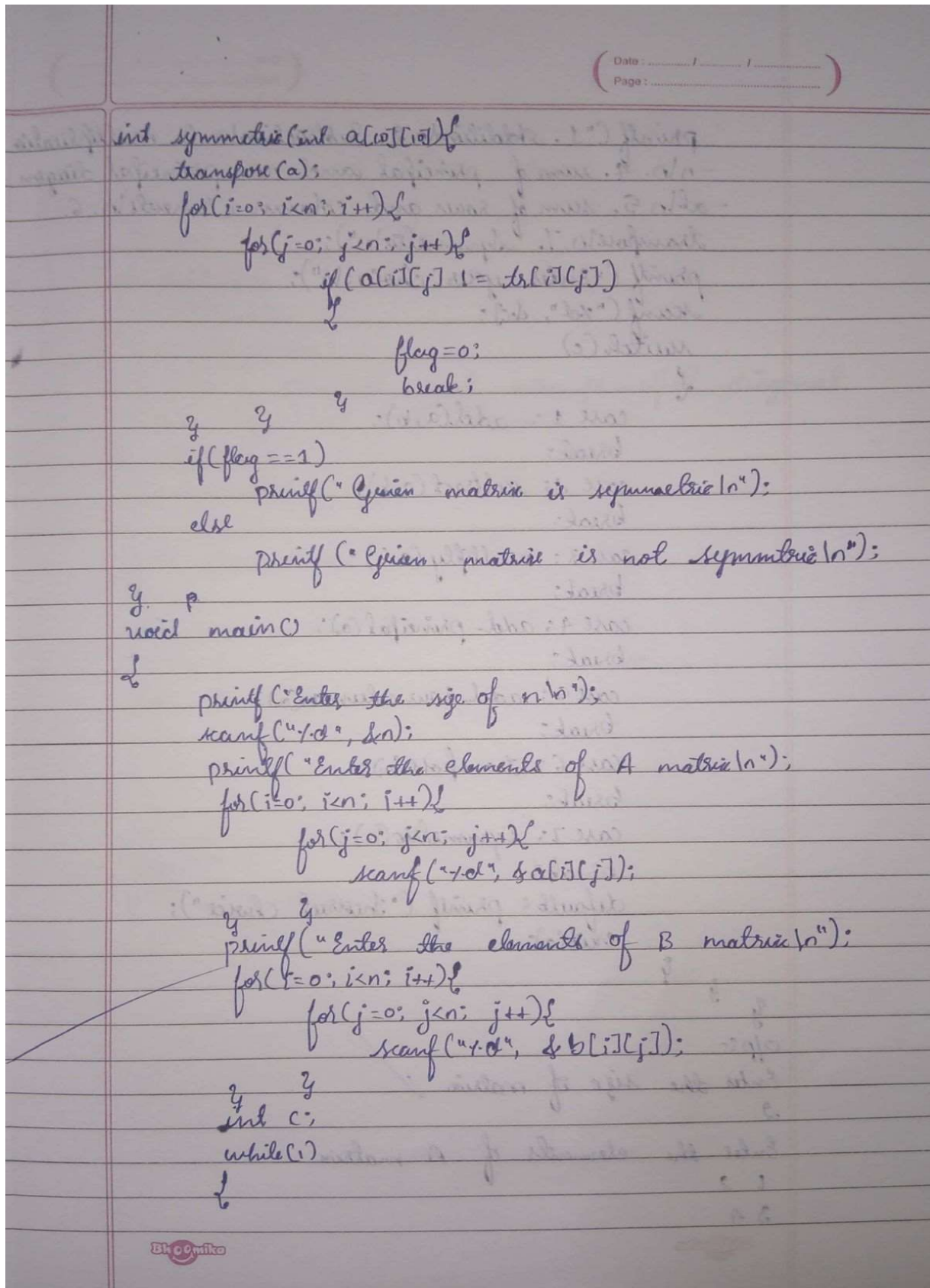
break;
printf("the sum of principal diagonal elements:\n");
printf("%d\n", psum);
printf("the sum of non principal diagonal elements:\n");
printf("%d\n", nonPsum);

int add_row_column(int a[10][10])
{
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            row += a[i][j];
    printf("The sum of %d row elements\n", i+1);
    for(j=0; j<n; j++)
        for(i=0; i<n; i++)
            column += a[i][j];
    printf("The sum of %d column elements\n", j+1);
}

int transpose(int a[10][10])
{
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            t[a[i][j]] = a[j][i];
    printf("the transpose of A matrix:\n");
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            printf("%d\t", t[i][j]);
    printf("\n");
}

```

Bhoomika





```

printf("1. Addition\n2. Subtraction\n3. multiplication\n4. sum of principal and non principal diagonal\n5. sum of row and column elements\n6. transpose\n7. symmetric\n");
printf("Enter your choice:\n");
scanf("%d", &c);
switch(c)
{
    case 1: add(a,b);
            break;
    case 2: subtract(a,b);
            break;
    case 3: multiply(a,b);
            break;
    case 4: add_principal(a);
            break;
    case 5: add_row_column(a);
            break;
    case 6: transpose(a);
            break;
    case 7: symmetric(a);
            break;
    default: printf("Incorrect choice");
            exit(0);
}
}
}

```

o/p:-

Enter the size of matrix

2

Enter the elements of A matrix

1 2

3 4

Bhaskar

( Date : \_\_\_\_\_ Page : \_\_\_\_\_ )

Enter the elements of B matrix

5 6  
5 0

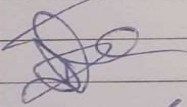
1. Addition  
2. Subtraction  
3. Multiplication  
4. Sum of principal and non principal diagonal  
5. Sum of row and column  
6. Transpose  
7. Symmetric

Enter your choice:

1.

The sum matrix is

5 8  
9 4

  
17-6-2023

Bhoomika

OUTPUT:

```
Enter the size of n
2
Enter the elements of A matrix
1 2
3 4
Enter the elements of B matrix
5 6
6 0
A matrix is
1      2
3      4
B matrix is
5      6
6      0
1.Addition
2.Subtraction
3.multiplication
4.sum of principal and non principal diagonal
5. sum of rows and columns
6.transpose
7.Symmetric
Enter your choice:
1
the sum matrix is:
6      8
9      4

1.Addition
2.Subtraction
3.multiplication
4.sum of principal and non principal diagonal
5. sum of rows and columns
6.transpose
7.Symmetric
Enter your choice:
2
the resultant matrix after subtraction is:
-4      -4
-3      4

1.Addition
2.Subtraction
3.multiplication
4.sum of principal and non principal diagonal
5. sum of rows and columns
6.transpose
7.Symmetric
Enter your choice:
3
the resultant matrix after multiplication is:
17      6
39      18

1.Addition
2.Subtraction
3.multiplication
4.sum of principal and non principal diagonal
5. sum of rows and columns
6.transpose
7.Symmetric
```

```

Enter your choice:
4
the sum of principal diagonal elements:
5
the sum of non principal diagonal elements:
5

1.Addition
2.Subtraction
3.multiplication
4.sum of principal and non principal diagonal
5. sum of rows and columns
6.transpose
7.Symmetric
Enter your choice:
5
The sum of rows is 10
the sum of column elements is 10
1.Addition
2.Subtraction
3.multiplication
4.sum of principal and non principal diagonal
5. sum of rows and columns
6.transpose
7.Symmetric
Enter your choice:
6
the transpose of A matrix:
1      3
2      4

1.Addition
2.Subtraction
3.multiplication
4.sum of principal and non principal diagonal
5. sum of rows and columns
6.transpose
7.Symmetric
Enter your choice:
7
the transpose of A matrix:
1      3
2      4

Given matrix is not symmetric

1.Addition
2.Subtraction
3.multiplication
4.sum of principal and non principal diagonal
5. sum of rows and columns
6.transpose
7.Symmetric
Enter your choice:
8
Incorrect choice
Process returned 0 (0x0)   execution time : 57.859 s
Press any key to continue.

```