

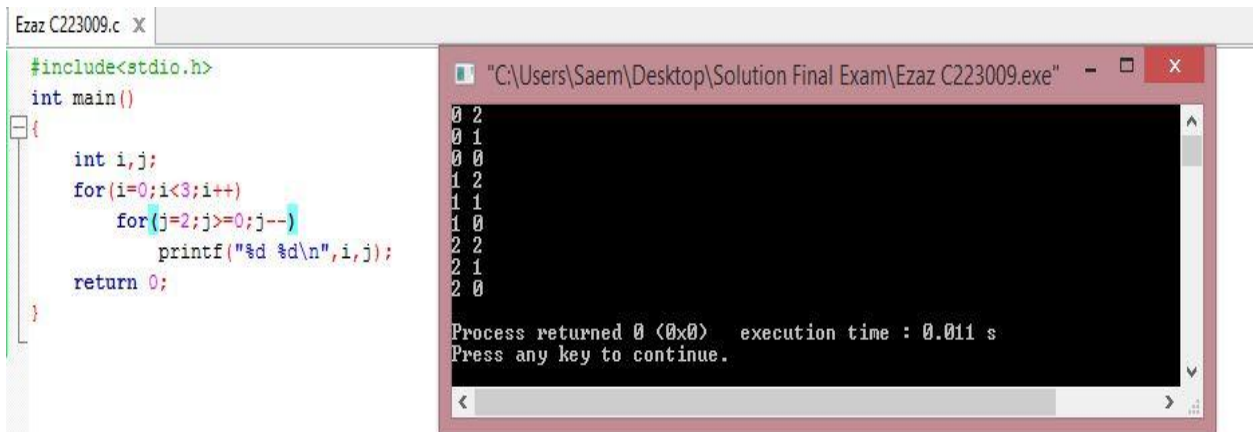
## Revised CSE Final Spring 22 Solution

Name:Ezaz Ahmed ID:C223009

এখানে কিছু কোডের দুইটা করে সলিউশন আছে একটা আমার আর বিকল্প সলিউশন আমার ফ্রেন্ড FahimulC223110 এর। যে যারটা ভালো করে বুঝা বা ওইটাই করবা।

1(a)

```
#include<stdio.h>
int main()
{
    int i,j;
    for(i=0;i<3;i++)
        for(j=2;j>=0;j--)
            printf("%d %d\n",i,j);
    return 0;
}
```

The screenshot shows a code editor window titled 'Ezaz C223009.c' on the left and a terminal window titled '"C:\Users\Saem\Desktop\Solution Final Exam\Ezaz C223009.exe"' on the right. The code in the editor is the same as the one in block 1(a). The terminal output shows the execution of the program, which prints a 3x3 grid of numbers from 0 to 2. The output is: 0 2, 0 1, 0 0, 1 2, 1 1, 1 0, 2 2, 2 1, 2 0. Below the output, the terminal shows 'Process returned 0 (0x0) execution time : 0.011 s' and 'Press any key to continue.'

```
#include<stdio.h>
int main()
{
    int i,j;
    for(i=0;i<3;i++)
        for(j=2;j>=0;j--)
            printf("%d %d\n",i,j);
    return 0;
}
```

```
0 2
0 1
0 0
1 2
1 1
1 0
2 2
2 1
2 0

Process returned 0 (0x0) execution time : 0.011 s
Press any key to continue.
```

1(b)

```
#include<stdio.h>
int main()
{
    int a=8,n=30,sum=0;
    int i;
    for(i=a;i<=n;i=i+3)
    {
        if(i%5==0)
        {
            sum=0;
            continue;
        }
        sum=sum+i;
        printf("sum = %d\n",sum);
    }
    return 0;}
}
```

```
Ezaz C223009.c X
#include<stdio.h>
int main()
{
    int a=8,n=30,sum=0;
    int i;
    for(i=a;i<=n;i=i+3)
    {
        if(i%5==0)
        {
            sum=0;
            continue;
        }
        sum=sum+i;
        printf("sum = %d\n",sum);
    }
    return 0;
}
```

```
"C:\Users\Saem\Desktop\Solution Final Exam\Ezaz C223009.exe... - [ ] [X]
sum = 8
sum = 19
sum = 33
sum = 50
sum = 23
sum = 49
sum = 78

Process returned 0 (0x0)   execution time : 0.010 s
Press any key to continue.
```

Again;

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

```
int main()
```

```
{
    int a=8,n=30,sum=0;
    int i;
    for(i=a;i<=n;i=i+3)
    {
        if(i%5==0)
        {
            sum=0;
            break;
        }
        sum=sum+i;
        printf("sum = %d\n",sum);
    }
    return 0;
}
```

```
Ezaz C223009.c X
#include<stdio.h>
int main()
{
    int a=8,n=30,sum=0;
    int i;
    for(i=a;i<=n;i=i+3)
    {
        if(i%5==0)
        {
            sum=0;
            break;
        }
        sum=sum+i;
        printf("sum = %d\n",sum);
    }
    return 0;
}
```

```
"C:\Users\Saem\Desktop\Solution Final Exam\Ezaz C223009.exe" - [ ] [X]
sum = 8
sum = 19
sum = 33
sum = 50

Process returned 0 (0x0)   execution time : 0.012 s
Press any key to continue.
```

1(c):

Solved By EzazC223009-

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

```
int main()
```

```
{
```

```
    int i, j, N, columns;
```

```
    columns=1;
```

```
    scanf("%d",&N);
```

```
    for(i=1; i<N*2; i++)
```

```
    {
```

```
        for(j=1; j<=columns; j++)
```

```
        {
```

```
            printf("*");
```

```
        }
```

```
        if(i < N)
```

```
        {
```

```
            columns++;
```

```
        }
```

```
        else
```

```
        {
```

```
            columns--;
```

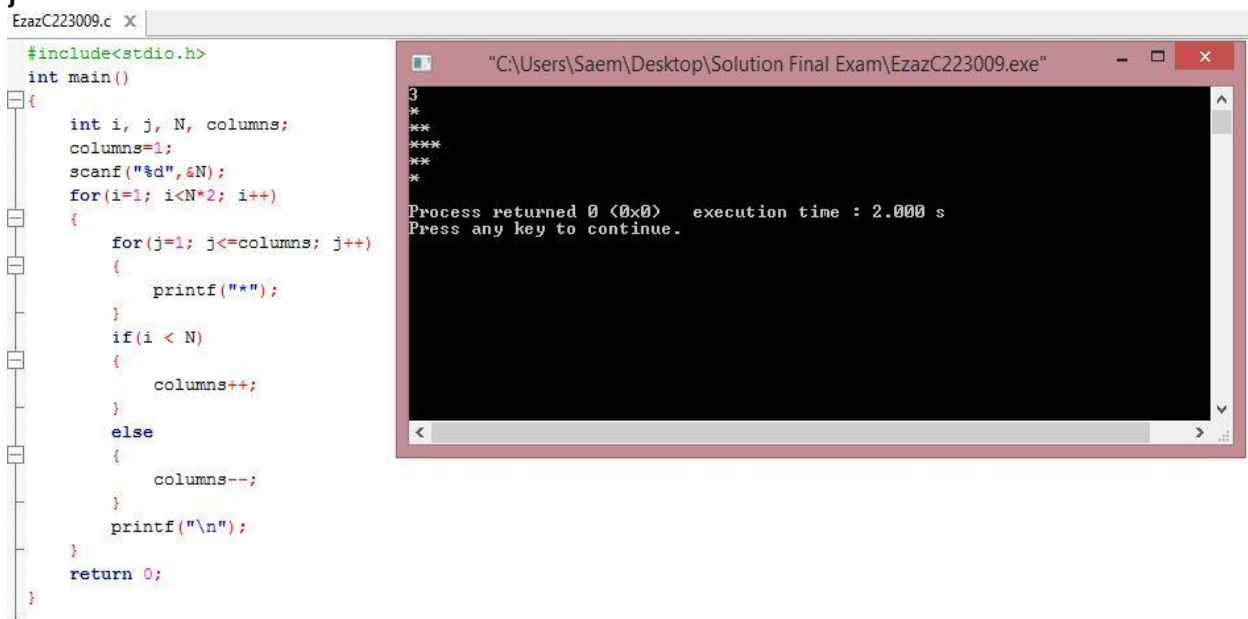
```
        }
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```



The screenshot shows a code editor on the left and a terminal window on the right. The code editor displays the C program code, and the terminal window shows the output of the program, which is a pattern of asterisks. The terminal window also displays the message "Process returned 0 (0x0) execution time : 2.000 s" and "Press any key to continue."

```
#include<stdio.h>
int main()
{
    int i, j, N, columns;
    columns=1;
    scanf("%d",&N);
    for(i=1; i<N*2; i++)
    {
        for(j=1; j<=columns; j++)
        {
            printf("*");
        }
        if(i < N)
        {
            columns++;
        }
        else
        {
            columns--;
```

```

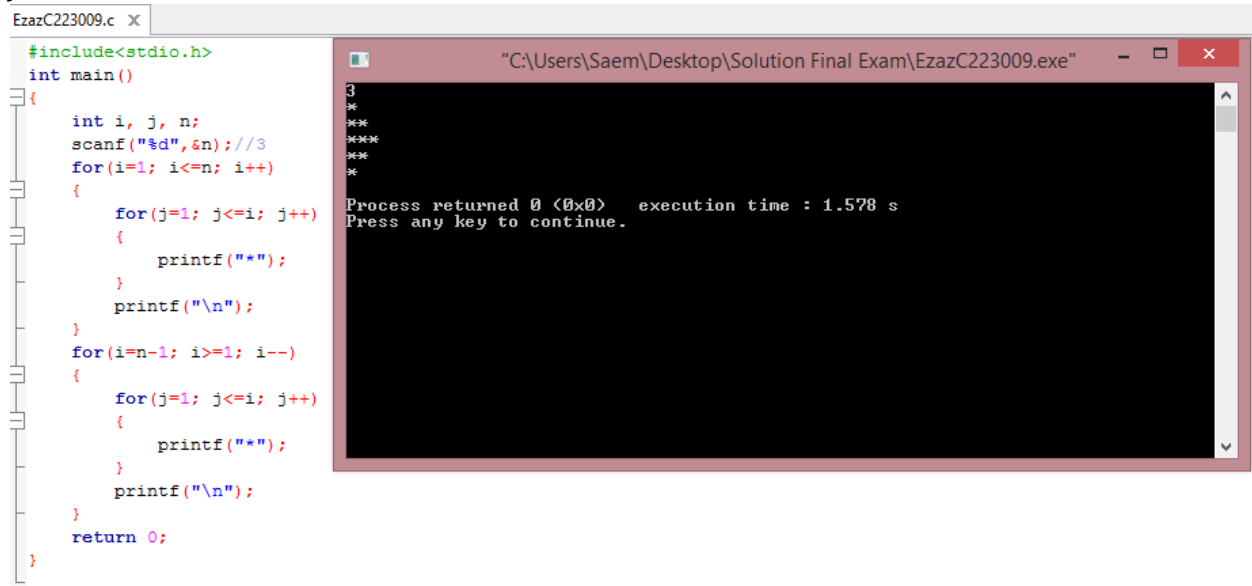
        printf("\n");
    }
    return 0;
}
```

3  
\*\*  
\*\*\*  
\*\*\*  
\*\*

Process returned 0 (0x0) execution time : 2.000 s  
Press any key to continue.

**Alternative Solution of 1(c) by FahimulC223110:**

```
#include<stdio.h>
int main()
{
    int i, j, n;
    scanf("%d",&n);//3
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=i; j++)
        {
            printf("*");
        }
        printf("\n");
    }
    for(i=n-1; i>=1; i--)
    {
        for(j=1; j<=i; j++)
        {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```



The screenshot displays a C program in a code editor and its execution in a command prompt window. The code editor shows the source code for a program that prints a diamond shape of asterisks. The command prompt window shows the output of the program, which is a diamond shape of asterisks. The output is as follows:

```
3
*
**
***
**
*
Process returned 0 (0x0)   execution time : 1.578 s
Press any key to continue.
```

1(d)

Solved By EzazC223009-

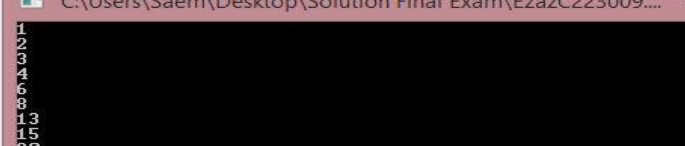
```
#include<stdio.h>
```

```
int main()
```

```
{  
    int a,i,sum=0,count=0;  
    for(i=0;;i++)  
    {  
        scanf("%d",&a);  
        if(a%3==0 || a%5==0)  
        {  
            continue;  
        }  
        else if(a<0)  
        {  
            break;  
        }  
        if(a%3!=0 || a%5!=0)  
        {  
            sum+=a;  
            count++;  
        }  
    }  
    double avg;  
    avg=(double)sum/count;  
    printf("Sum: %d\n",sum);  
    printf("Average: %.2lf\n",avg);  
    return 0;  
}
```

```
EzazC223009.c x
#include<stdio.h>
int main()
{
    int a,i,sum=0,count=0;
    for(i=0;;i++)
    {
        scanf("%d",&a);
        if(a%3==0 || a%5==0)
        {
            continue;
        }
        else if(a<0)
        {
            break;
        }
        if(a%3!=0 || a%5!=0)
        {
            sum+=a;
            count++;
        }
    }

    double avg;
    avg=(double)sum/count;
    printf("Sum: %d\n",sum);
    printf("Average: %.2lf\n",avg);
    return 0;
}
```



1(d) or;

Solved By EzazC223009-

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

```
int main()
```

```
{
```

```
    int n, i, flag = 0;
```

```
    printf("Enter a positive integer: ");
```

```
    scanf("%d", &n);
```

```
    if (n == 0 || n == 1)
```

```
    {
```

```
        flag = 1;
```

```
    }
```

```
    for (i = 2; i <= n / 2; ++i)
```

```
    {
```

```
        if (n % i == 0)
```

```
        {
```

```
            flag = 1;
```

```
            break;
```

```
        }
```

```
    }
```

```
    if (flag == 0)
```

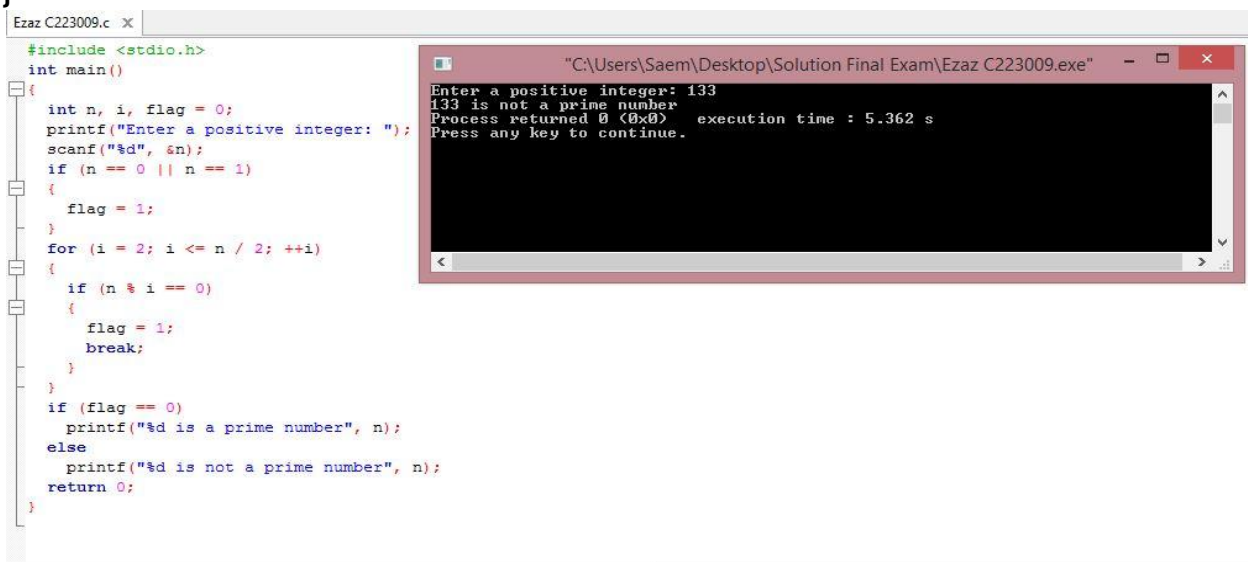
```
        printf("%d is a prime number", n);
```

```
    else
```

```
        printf("%d is not a prime number", n);
```

```
    return 0;
```

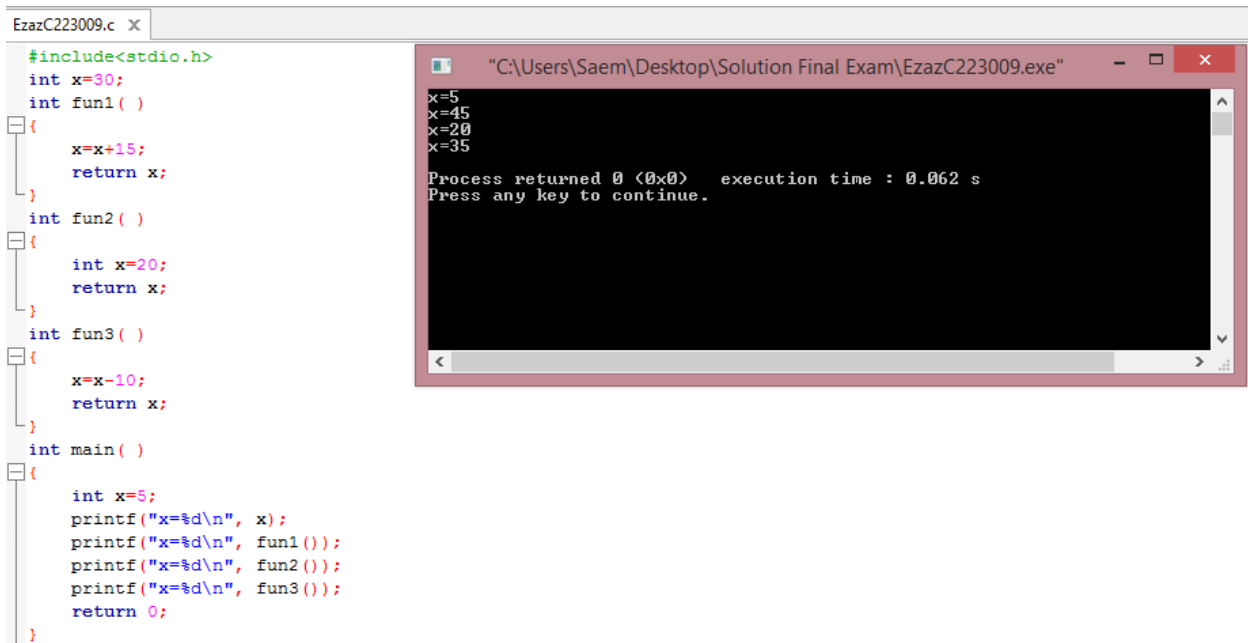
```
}
```



```
Ezaz C223009.c x
#include <stdio.h>
int main()
{
    int n, i, flag = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    if (n == 0 || n == 1)
    {
        flag = 1;
    }
    for (i = 2; i <= n / 2; ++i)
    {
        if (n % i == 0)
        {
            flag = 1;
            break;
        }
    }
    if (flag == 0)
        printf("%d is a prime number", n);
    else
        printf("%d is not a prime number", n);
    return 0;
}
```

```
"C:\Users\Saem\Desktop\Solution Final Exam\Ezaz C223009.exe"
Enter a positive integer: 133
133 is not a prime number
Process returned 0 (0x0)   execution time : 5.362 s
Press any key to continue.
```

2(b)i:



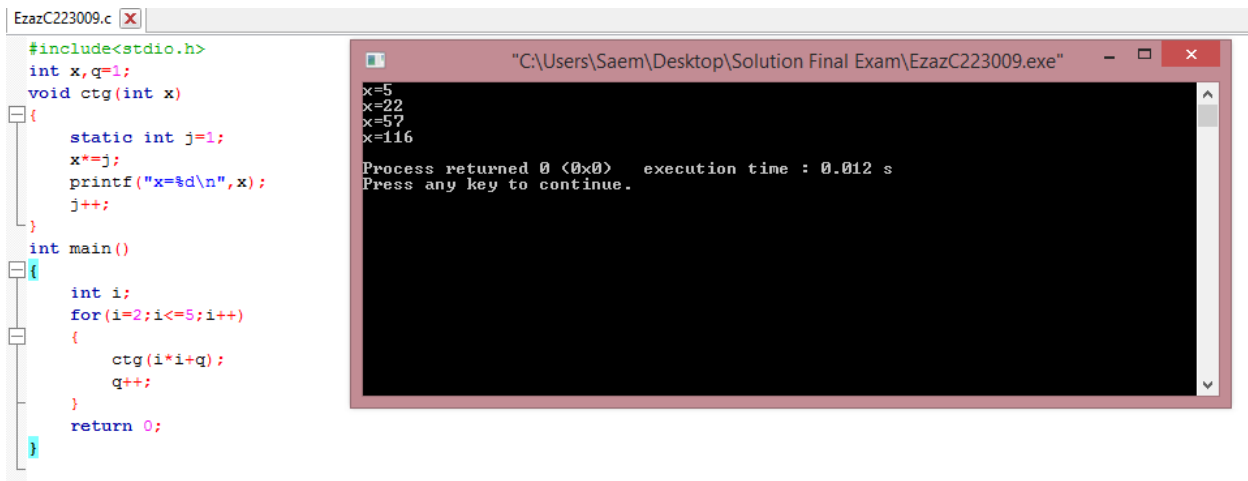
The screenshot shows a C program in a text editor and its execution in a command prompt window. The program defines three functions: fun1, fun2, and fun3, each taking an integer and returning a value. The main function calls these functions and prints the results.

```
#include<stdio.h>
int x=30;
int fun1 ( )
{
    x=x+15;
    return x;
}
int fun2 ( )
{
    int x=20;
    return x;
}
int fun3 ( )
{
    x=x-10;
    return x;
}
int main ( )
{
    int x=5;
    printf("x=%d\n", x);
    printf("x=%d\n", fun1 ());
    printf("x=%d\n", fun2 ());
    printf("x=%d\n", fun3 ());
    return 0;
}
```

The execution output shows the values of x at each step:

```
x=5
x=45
x=20
x=35
Process returned 0 (0x0)   execution time : 0.062 s
Press any key to continue.
```

2(b)ii:



The screenshot shows a C program in a text editor and its execution in a command prompt window. The program defines a function ctg that takes an integer x and returns a value. The main function calls this function in a loop and prints the results.

```
#include<stdio.h>
int x,q=1;
void ctg(int x)
{
    static int j=1;
    x*=j;
    printf("x=%d\n",x);
    j++;
}
int main()
{
    int i;
    for(i=2;i<=5;i++)
    {
        ctg(i*i+q);
        q++;
    }
    return 0;
}
```

The execution output shows the values of x at each step:

```
x=5
x=22
x=57
x=116
Process returned 0 (0x0)   execution time : 0.012 s
Press any key to continue.
```

2(d)

Solved By EzazC223009-

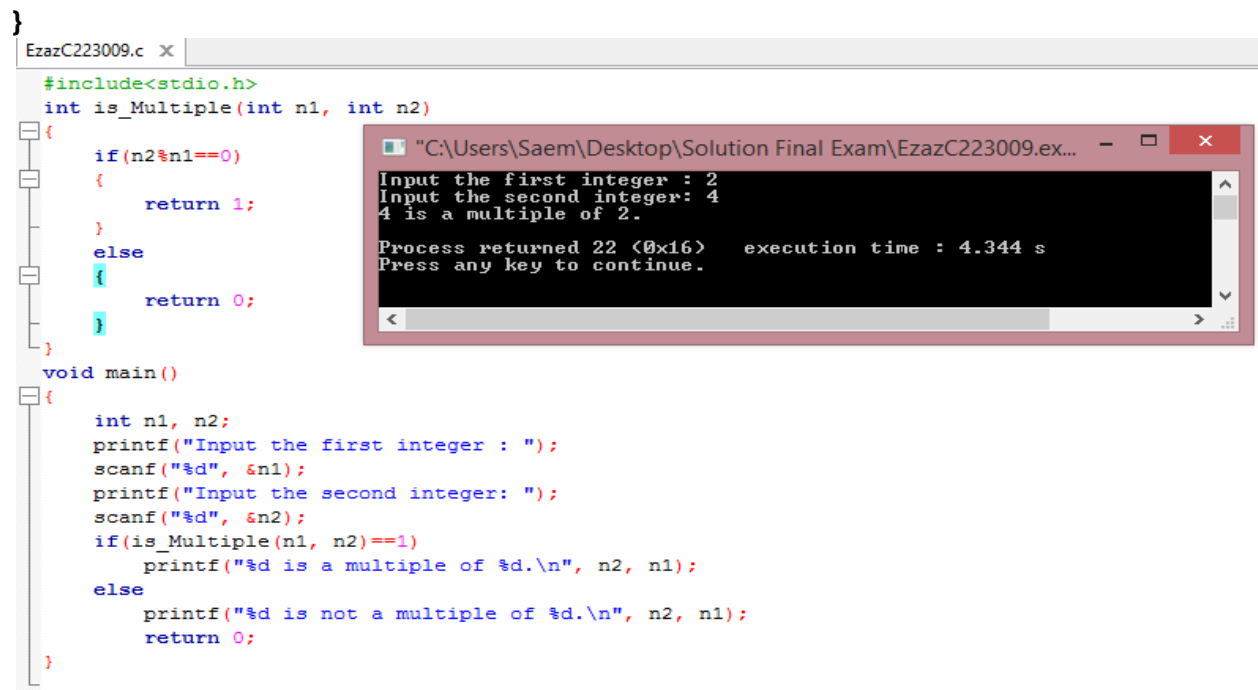
```
#include<stdio.h>
```

```
int is_Multiple(int n1, int n2)
```

```
{
    if(n2%n1==0)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
```

```
void main()
```

```
{
    int n1, n2;
    printf("Input the first integer : ");
    scanf("%d", &n1);
    printf("Input the second integer: ");
    scanf("%d", &n2);
    if(is_Multiple(n1, n2)==1)
        printf("%d is a multiple of %d.\n", n2, n1);
    else
        printf("%d is not a multiple of %d.\n", n2, n1);
    return 0;
}
```



The image shows a screenshot of a C program and its execution. On the left, the source code is displayed in a text editor window titled 'EzazC223009.c'. The code defines a function 'is\_Multiple' that checks if a number 'n2' is a multiple of 'n1' by using the modulo operator. The 'main' function prompts the user for two integers, reads them, and then calls 'is\_Multiple' to check if the second integer is a multiple of the first. On the right, a separate window shows the program's execution. It displays the prompts and user input: 'Input the first integer : 2' and 'Input the second integer: 4'. The output shows '4 is a multiple of 2.' followed by 'Process returned 22 (0x16) execution time : 4.344 s' and 'Press any key to continue.'

```
#include<stdio.h>
int is_Multiple(int n1, int n2)
{
    if(n2%n1==0)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

void main()
{
    int n1, n2;
    printf("Input the first integer : ");
    scanf("%d", &n1);
    printf("Input the second integer: ");
    scanf("%d", &n2);
    if(is_Multiple(n1, n2)==1)
        printf("%d is a multiple of %d.\n", n2, n1);
    else
        printf("%d is not a multiple of %d.\n", n2, n1);
    return 0;
}
```

"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.ex... - x

Input the first integer : 2  
Input the second integer: 4  
4 is a multiple of 2.

Process returned 22 (0x16) execution time : 4.344 s  
Press any key to continue.

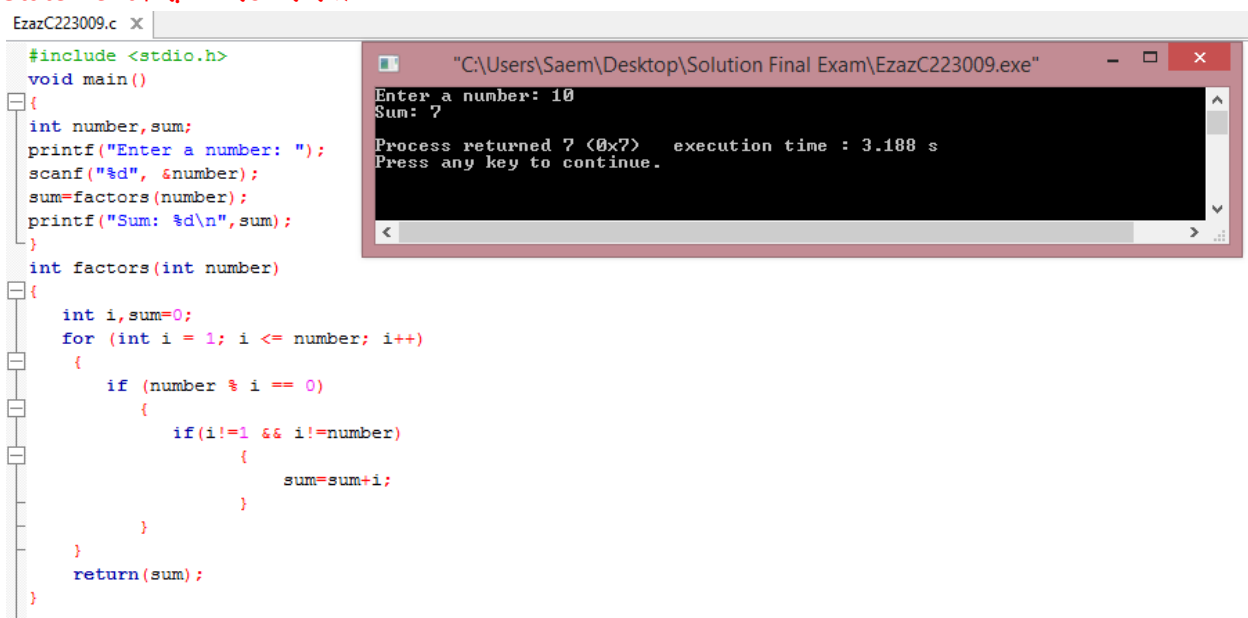


2(d) or:

Solved By EzazC223009-

```
#include <stdio.h>
void main()
{
    int number,sum;
    printf("Enter a number: ");
    scanf("%d", &number);
    sum=factors(number);
    printf("Sum: %d\n",sum);
}
int factors(int number)
{
    int i,sum=0;
    for (int i = 1; i <= number; i++)
    {
        if (number % i == 0)
        {
            if(i!=1 && i!=number)
            {
                sum=sum+i;
            }
        }
    }
    return(sum);
}
```

আমি এর আগের পিডিএফ এ Return Statement এর ব্যবহার ছাড়াই কোডটি সলভ করেছিলাম। বাট প্রশ্নে Return Statement দিয়ে করতে বলেছে।



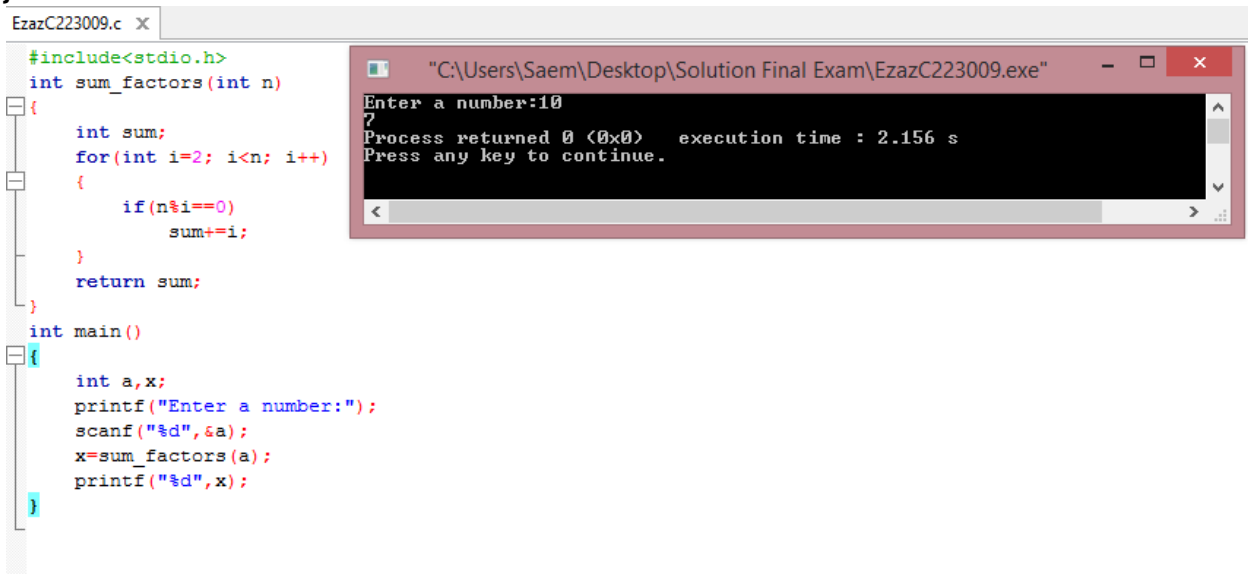
The screenshot shows a C program in a text editor and its execution in a command prompt window. The program calculates the sum of factors of a given number, excluding 1 and the number itself. The user entered 10, and the program outputted a sum of 7. The command prompt window also displays the process return code and execution time.

```
EzazC223009.c x
#include <stdio.h>
void main()
{
    int number,sum;
    printf("Enter a number: ");
    scanf("%d", &number);
    sum=factors(number);
    printf("Sum: %d\n",sum);
}
int factors(int number)
{
    int i,sum=0;
    for (int i = 1; i <= number; i++)
    {
        if (number % i == 0)
        {
            if(i!=1 && i!=number)
            {
                sum=sum+i;
            }
        }
    }
    return(sum);
}
```

```
"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"
Enter a number: 10
Sum: 7
Process returned 7 (0x7)   execution time : 3.188 s
Press any key to continue.
```

**Alternative Solution of 2(d)or By FahimulC223110:**

```
#include<stdio.h>
int sum_factors(int n)
{
    int sum;
    for(int i=2; i<n; i++)
    {
        if(n%i==0)
            sum+=i;
    }
    return sum;
}
int main()
{
    int a,x;
    printf("Enter a number:");
    scanf("%d",&a);
    x=sum_factors(a);
    printf("%d",x);
}
```



The image shows a C program in a text editor and its execution in a console window. The code defines a function `sum_factors` that calculates the sum of factors of a number `n` (excluding `n` itself). The `main` function prompts the user to enter a number, reads it, and prints the result of `sum_factors`.

**Code Snippet:**

```
#include<stdio.h>
int sum_factors(int n)
{
    int sum;
    for(int i=2; i<n; i++)
    {
        if(n%i==0)
            sum+=i;
    }
    return sum;
}
int main()
{
    int a,x;
    printf("Enter a number:");
    scanf("%d",&a);
    x=sum_factors(a);
    printf("%d",x);
}
```

**Execution Output:**

```
"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"
Enter a number:10
7
Process returned 0 (0x0)   execution time : 2.156 s
Press any key to continue.
```

3(a)

Solved By EzazC223009-

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

```
int main()
```

```
{
```

```
    int i;
```

```
    int arr[20]={2,2,3,0,0,9}; //My id=C223009
```

```
    for (int i = 1; i < 7; i++)
```

```
    {
```

```
        arr[i] += arr[i - 1];
```

```
    }
```

```
    printf("Cumulative sum of the array : ");
```

```
    for (i = 0; i < 6; i++)
```

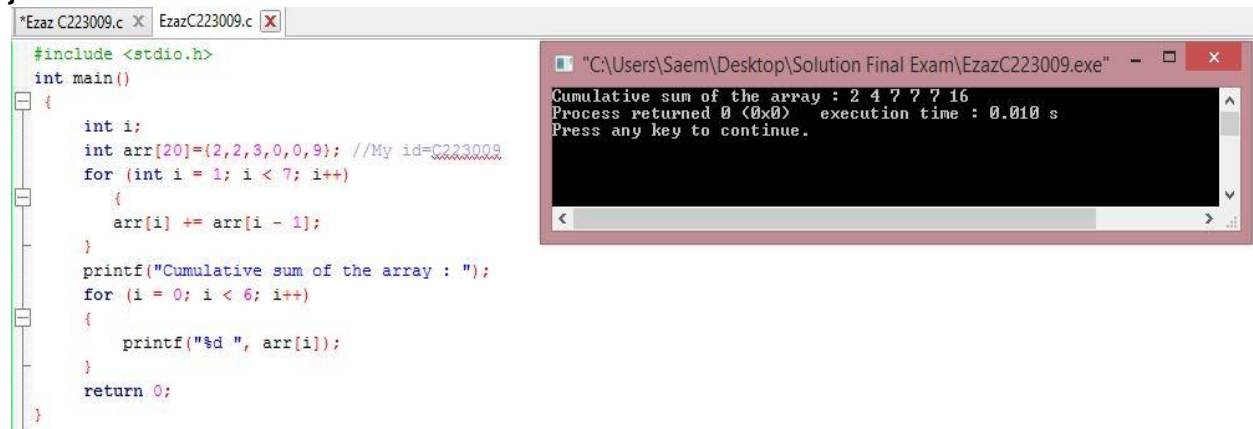
```
    {
```

```
        printf("%d ", arr[i]);
```

```
    }
```

```
    return 0;
```

```
}
```



The image shows a screenshot of a C program and its execution. On the left, a code editor window titled "Ezaz C223009.c" displays the following code:

```
#include <stdio.h>
int main()
{
    int i;
    int arr[20]={2,2,3,0,0,9}; //My id=C223009
    for (int i = 1; i < 7; i++)
    {
        arr[i] += arr[i - 1];
    }
    printf("Cumulative sum of the array : ");
    for (i = 0; i < 6; i++)
    {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

On the right, a console window titled "\"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe\"" shows the output of the program:

```
Cumulative sum of the array : 2 4 7 7 7 16
Process returned 0 (0x0)   execution time : 0.010 s
Press any key to continue.
```

**Go to the next page**



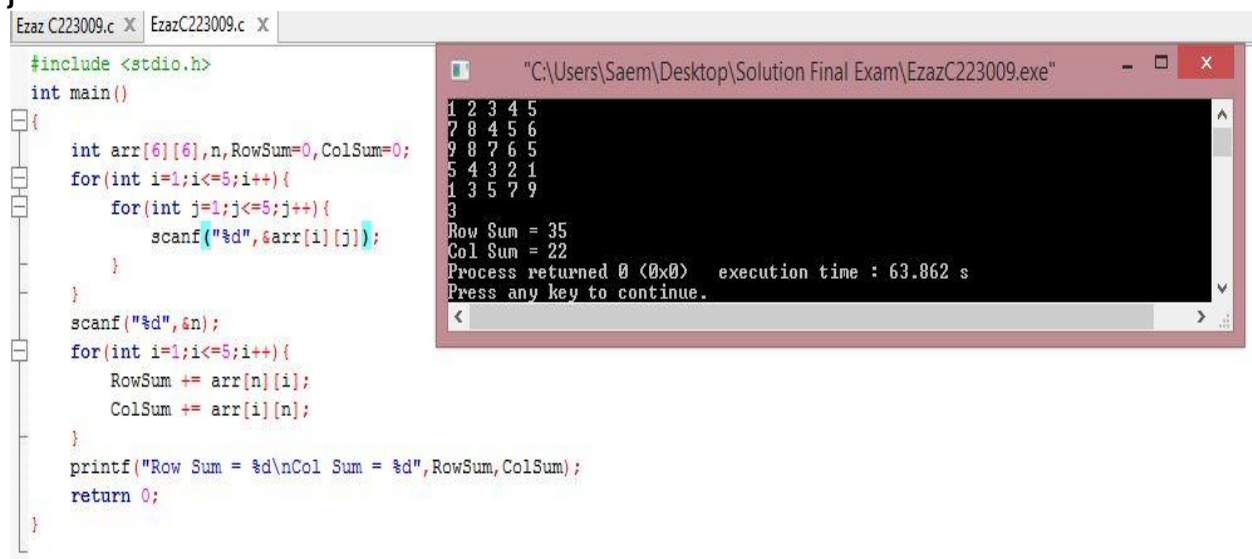
3(b):

**Solved By EzazC223009-**

**#include <stdio.h>**

**int main()**

```
{  
    int arr[6][6],n,RowSum=0,ColSum=0;  
    for(int i=1;i<=5;i++){  
        for(int j=1;j<=5;j++){  
            scanf("%d",&arr[i][j]);  
        }  
    }  
    scanf("%d",&n);  
    for(int i=1;i<=5;i++){  
        RowSum += arr[n][i];  
        ColSum += arr[i][n];  
    }  
    printf("Row Sum = %d\nCol Sum = %d",RowSum,ColSum);  
    return 0;  
}
```



```
#include <stdio.h>  
int main()  
{  
    int arr[6][6],n,RowSum=0,ColSum=0;  
    for(int i=1;i<=5;i++){  
        for(int j=1;j<=5;j++){  
            scanf("%d",&arr[i][j]);  
        }  
    }  
    scanf("%d",&n);  
    for(int i=1;i<=5;i++){  
        RowSum += arr[n][i];  
        ColSum += arr[i][n];  
    }  
    printf("Row Sum = %d\nCol Sum = %d",RowSum,ColSum);  
    return 0;  
}
```

"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"

```
1 2 3 4 5  
7 8 4 5 6  
9 8 7 6 5  
5 4 3 2 1  
1 3 5 7 9  
3  
Row Sum = 35  
Col Sum = 22  
Process returned 0 (0x0)   execution time : 63.862 s  
Press any key to continue.
```

**Go to the next page**



3(b) or:

Solved By EzazC223009-

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

```
int main()
```

```
{
```

```
    int array[100][100];
```

```
    int i, j, m, n, a = 0, sum = 0;
```

```
    printf("Enter the order of the matrix \n");
```

```
    scanf("%d %d", &m, &n);
```

```
    if (m == n )
```

```
    {
```

```
        printf("Enter the co-efficients of the matrix\n");
```

```
        for (i = 0; i < m; ++i)
```

```
        {
```

```
            for (j = 0; j < n; ++j)
```

```
            {
```

```
                scanf("%d", &array[i][j]);
```

```
            }
```

```
        }
```

```
        for (i = 0; i < m; ++i)
```

```
        {
```

```
            sum = sum + array[i][i];
```

```
            a = a + array[i][m - i - 1];
```

```
        }
```

```
        printf("\nThe sum of the main diagonal elements is = %d\n", sum);
```

```
        printf("The sum of the secondary diagonal elements is = %d\n", a);
```

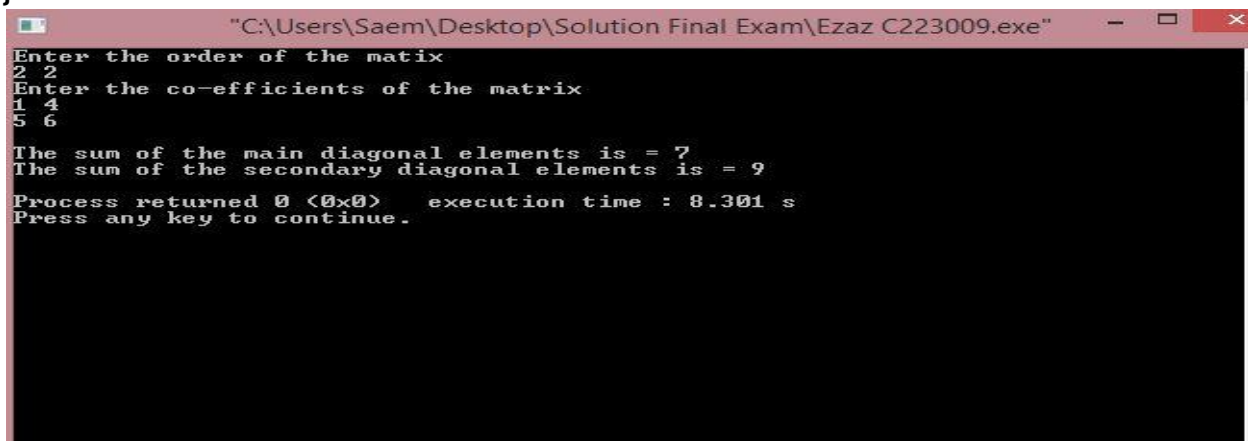
```
    }
```

```
    else
```

```
        printf("The given order is not square matrix\n");
```

```
    return 0;
```

```
}
```



```
"C:\Users\Saem\Desktop\Solution Final Exam\Ezaz C223009.exe"
Enter the order of the matrix
2 2
Enter the co-efficients of the matrix
1 4
5 6

The sum of the main diagonal elements is = 7
The sum of the secondary diagonal elements is = 9

Process returned 0 (0x0)   execution time : 8.301 s
Press any key to continue.
```

Alternative Solution of 3(b)or By FahimulC223110:

```
#include <stdio.h>
#include<math.h>
int main()
{
    int i, j, n, k, sum1 = 0,sum2=0;
    scanf("%d", &n);
    int a[n + 5][n + 5];
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            if (i == j)
            {
                sum1 += a[i][j];
            }
        }
    }
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            if (i+j == n-1)
            {
                sum2 += a[i][j];
            }
        }
    }
    printf("sum of the elements of 1st diagonal = %d\n",sum1);
    printf("sum of the elements of 2nd diagonal = %d\n",sum2);
    return 0;
}
```

```
"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"
2
1 4
5 6
sum of the elements of 1st diagonal = 7
sum of the elements of 2nd diagonal = 9

Process returned 0 (0x0)   execution time : 7.297 s
Press any key to continue.
```

3(c):

```
EzazC223009.c x
#include<stdio.h>
#include<string.h>
int main()
{
    char s1[8]="IIUC",s2[8]="CSE",s3[8]="1121";
    printf("%d\n",strlen(s1));
    printf("%s\n",strcpy(s1,s2));
    printf("%s\n",strcat(s2,s3));
    printf("%s\n",strrev(s2));
    return 0;
}
```

```
"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"
4
CSE
CSE1121
1211ESC

Process returned 0 (0x0)   execution time : 0.045 s
Press any key to continue.
```

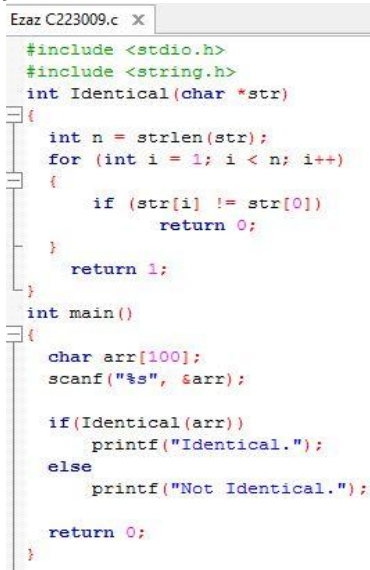
3(d)

Solved By EzazC223009-

```
#include <stdio.h>
#include <string.h>
int Identical(char *str)
{
    int n = strlen(str);
    for (int i = 1; i < n; i++)
    {
        if (str[i] != str[0])
            return 0;
    }
    return 1;
}
int main()
{
    char arr[100];
    scanf("%s", &arr);

    if(Identical(arr))
        printf("Identical.");
    else
        printf("Not Identical.");

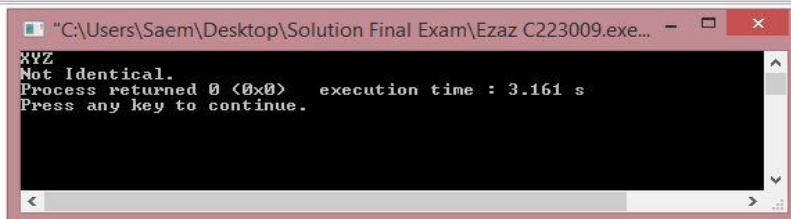
    return 0;
}
```

A screenshot of a code editor window titled "Ezaz C223009.c". The code is the same as the one in the previous block, but with syntax highlighting. The code defines a function 'Identical' that checks if all characters in a string are the same as the first character. The 'main' function reads a string 'arr' and prints "Identical." or "Not Identical." based on the function's result.

```
#include <stdio.h>
#include <string.h>
int Identical(char *str)
{
    int n = strlen(str);
    for (int i = 1; i < n; i++)
    {
        if (str[i] != str[0])
            return 0;
    }
    return 1;
}
int main()
{
    char arr[100];
    scanf("%s", &arr);

    if(Identical(arr))
        printf("Identical.");
    else
        printf("Not Identical.");

    return 0;
}
```

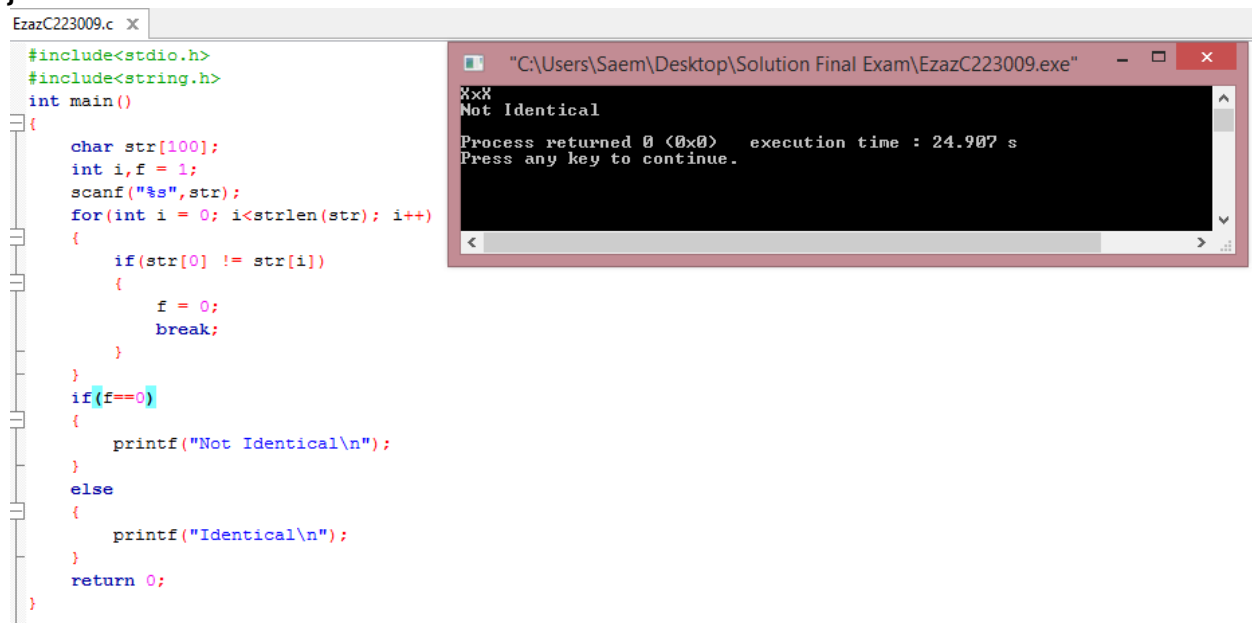
A screenshot of a command prompt window titled "C:\Users\Saem\Desktop\Solution Final Exam\Ezaz C223009.exe...". The output shows the program's execution: it prints "XYZ", then "Not Identical.", followed by "Process returned 0 (0x0) execution time : 3.161 s" and "Press any key to continue.".

```
XYZ
Not Identical.
Process returned 0 (0x0) execution time : 3.161 s
Press any key to continue.
```



**Alternative Solution of 3(d) By FahimulC223110:**

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[100];
    int i,f = 1;
    scanf("%s",str);
    for(int i = 0; i<strlen(str); i++)
    {
        if(str[0] != str[i])
        {
            f = 0;
            break;
        }
    }
    if(f==0)
    {
        printf("Not Identical\n");
    }
    else
    {
        printf("Identical\n");
    }
    return 0;
}
```



The screenshot displays a C program in a code editor and its execution output in a separate window. The code editor on the left shows the source code for a program that checks if a string is identical to itself. The output window on the right shows the program's execution, including the input 'xxx', the output 'Not Identical', and the execution time of 24.987 seconds.

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[100];
    int i,f = 1;
    scanf("%s",str);
    for(int i = 0; i<strlen(str); i++)
    {
        if(str[0] != str[i])
        {
            f = 0;
            break;
        }
    }
    if(f==0)
    {
        printf("Not Identical\n");
    }
    else
    {
        printf("Identical\n");
    }
    return 0;
}
```

Output Window: "C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"

```
xxx
Not Identical
Process returned 0 (0x0)   execution time : 24.987 s
Press any key to continue.
```

3(d) or;

Solved By EzazC223009-

#include<stdio.h>

int main()

```
{
    char s[100];
    scanf("%s",&s);
    if(s[2]=='T' && s[3]=='N')
    {
        printf("Age=%c%c,Taxpayer,Not a Landowner",s[0],s[1]);
    }
    else if(s[2]=='T' && s[3]=='L')
    {
        printf("Age=%c%c,Taxpayer,Landowner",s[0],s[1]);
    }
    else if(s[2]=='N' && s[3]=='N')
    {
        printf("Age=%c%c,Not a Taxpayer,Not a Landowner",s[0],s[1]);
    }
    else if(s[2]=='N' && s[3]=='L')
    {
        printf("Age=%c%c,Not a Taxpayer,Landowner",s[0],s[1]);
    }
    return 0;
}
```



The image shows a screenshot of a C program being edited in a text editor and its execution output in a command prompt window.

The C program code is as follows:

```
#include<stdio.h>
int main()
{
    char s[100];
    scanf("%s",&s);
    if(s[2]=='T' && s[3]=='N')
    {
        printf("Age=%c%c,Taxpayer,Not a Landowner",s[0],s[1]);
    }
    else if(s[2]=='T' && s[3]=='L')
    {
        printf("Age=%c%c,Taxpayer,Landowner",s[0],s[1]);
    }
    else if(s[2]=='N' && s[3]=='N')
    {
        printf("Age=%c%c,Not a Taxpayer,Not a Landowner",s[0],s[1]);
    }
    else if(s[2]=='N' && s[3]=='L')
    {
        printf("Age=%c%c,Not a Taxpayer,Landowner",s[0],s[1]);
    }
    return 0;
}
```

The command prompt window shows the execution of the program. The user has entered "43TN" as input. The output of the program is "Age=43,Taxpayer,Not a Landowner". The command prompt also shows the process returned 0 (0x0) and the execution time was 4.966 s. The prompt asks to press any key to continue.

**4(b):** The **main difference** between pass by value and pass by reference is that, **in a pass by value, the parameter value copies to another variable while, in a pass by reference, the actual parameter passes to the function.**

Example:

Pass by Value:

```
#include <stdio.h>
void swap (int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
int main ()
{
    int a = 10;
    int b = 20;
    printf ("Before swap, a = %d, b = %d\n", a, b);
    swap (a, b);
    printf ("After swap, a = %d, b = %d\n", a, b);
    return 0;
}
```

Pass By Reference:

```
#include <stdio.h>
int main ()
{
    int a = 100;
    int b = 200;
    printf("Before swap, value of a : %d\n", a);
    printf("Before swap, value of b : %d\n", b);
    swap(&a, &b);
    printf("After swap, value of a : %d\n", a);
    printf("After swap, value of b : %d\n", b);
    return 0;
}
void swap(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
    return;
}
```

4(d)

Solved By EzazC223009-

```
#include <stdio.h>
int main()
{
    struct Player
    {
        int runs;
        char name[24],country[16];
        double average;
    }
    player[10];
    int i;
    for(i=0;i<10;i++)
    {
        printf("Enter Name: ");
        scanf("%s",player[i].name);
        printf("Enter Country: ");
        scanf("%s",&player[i].country);
        printf("Enter Runs: ");
        scanf("%d",&player[i].runs);
        printf("Enter Average: ");
        scanf("%lf",&player[i].average);
    }
    for(i=0;i<10;i++)
    {
        printf("\nName: %s",player[i].name);
        printf("\nCountry: %s",player[i].country);
        printf("\nRuns: %d",player[i].runs);
        printf("\nAverage: %.3lf",player[i].average);
    }
    int max;
    max=0;
    for(i=0;i<10;i++)
    {
        if(player[i].average>player[max].average)
        {
            max=i;
        }
    }
    printf("\n\nThe player who has the highest average is: %s\n",player[max].name);
    return 0;
}
```

```
"C:\Users\Saem\Desktop\Solution Final Exam\Ezaz C223009.exe"
Enter Name: Ezaz
Enter Country: Bangladesh
Enter Runs: 23
Enter Average: 12.5
Enter Name: Wasee
Enter Country: Bangladesh
Enter Runs: 20
Enter Average: 9.89
Enter Name: John
Enter Country: England
Enter Runs: 15
Enter Average: 7.5

Name: Ezaz
Country: Bangladesh
Runs: 23
Average: 12.500
Name: Wasee
Country: Bangladesh
Runs: 20
Average: 9.890
Name: John
Country: England
Runs: 15
Average: 7.500

The player who has the highest average is: Ezaz

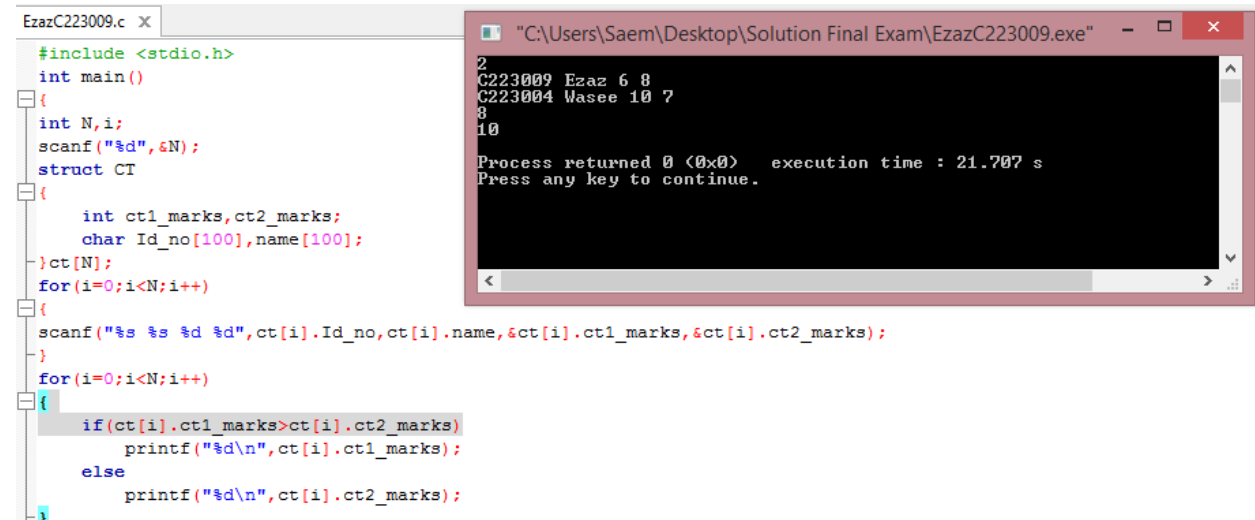
Process returned 0 (0x0)   execution time : 65.986 s
Press any key to continue.
```

আমি এখানে আউটপুট টা দেখানোর জন্য শুধু ৩টা ইনপুট নিয়েছি। বাকি কোডে ১০ জনের ইনপুট নিয়েছি।

**Go to the Next Page:**



4(d) or:



The image shows a C program in a text editor and its execution in a command prompt window. The program defines a struct CT with fields ct1\_marks, ct2\_marks, Id\_no, and name. It reads N, then N pairs of (Id\_no, name, ct1\_marks, ct2\_marks). For each pair, it prints the maximum of ct1\_marks and ct2\_marks.

```
#include <stdio.h>
int main()
{
    int N,i;
    scanf("%d",&N);
    struct CT
    {
        int ct1_marks,ct2_marks;
        char Id_no[100],name[100];
    }ct[N];
    for(i=0;i<N;i++)
    {
        scanf("%s %s %d %d",ct[i].Id_no,ct[i].name,&ct[i].ct1_marks,&ct[i].ct2_marks);
    }
    for(i=0;i<N;i++)
    {
        if(ct[i].ct1_marks>ct[i].ct2_marks)
            printf("%d\n",ct[i].ct1_marks);
        else
            printf("%d\n",ct[i].ct2_marks);
    }
}
```

Execution output:

```
C223009 Ezaz 6 8
C223004 Wasee 10 7
8
10
Process returned 0 (0x0)   execution time : 21.707 s
Press any key to continue.
```

```
#include <stdio.h>
int main()
{
    int N,i;
    scanf("%d",&N);
    struct CT
    {
        int ct1_marks,ct2_marks;
        char Id_no[100],name[100];
    }ct[N];
    for(i=0;i<N;i++)
    {
        scanf("%s %s %d %d",ct[i].Id_no,ct[i].name,&ct[i].ct1_marks,&ct[i].ct2_marks);
    }
    for(i=0;i<N;i++)
    {
        if(ct[i].ct1_marks>ct[i].ct2_marks)
            printf("%d\n",ct[i].ct1_marks);
        else
            printf("%d\n",ct[i].ct2_marks);
    }
    return 0;
}
```

5(a):

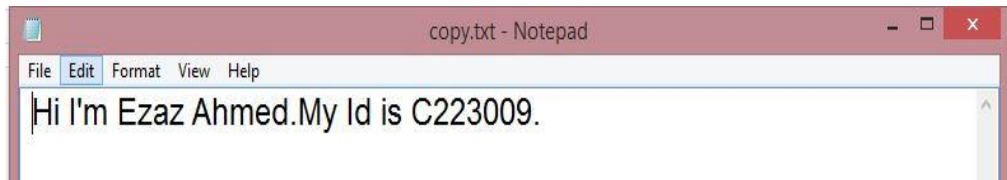
**feof()** function is used to determine if the end of the file (stream) specified has been reached or not. This function keeps on searching the end of the file (EOF) in the file program.

5(b):

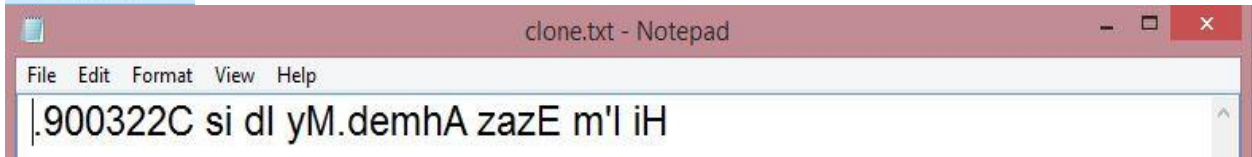
Solved By EzazC223009-

যদি প্রশ্নে **strrev()** use না করে কোড সলভ করতে বলে তাহলে এভাবে করতে হবে:

```
#include<stdio.h>
int main()
{
    FILE *fp;
    FILE *rev;
    char ch;
    int i=0,pos;
    fp=fopen("copy.txt","r");
    rev=fopen("clone.txt","w");
    if(fp==NULL)
    {
        printf("File does not exist..");
        return 0;
    }
    fseek(fp,0,SEEK_END);
    pos=ftell(fp);
    while(i<pos)
    {
        i++;
        fseek(fp,-i,SEEK_END);
        ch=fgetc(fp);
        fputc(ch,rev);
    }
    fclose(fp);
    fclose(rev);
    return 0;
}
```



**After executing the code**



**Go to the Next Page:**

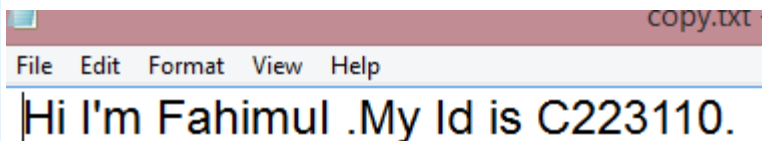




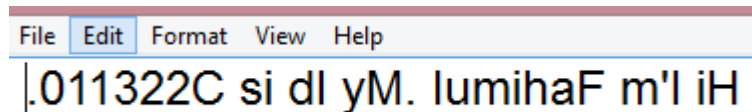
Alternative Solution of 5(b) By FahimulC223110:

যদি প্রস্নে strrev() ও use করা যাবে থাকে:

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[100],clone[100];
    FILE *fr;
    fr = fopen("copy.txt", "r");
    if(fr==NULL)
    {
        printf("File does not exist..");
        return 0;
    }
    fgets(str,100,fr);
    fclose(fr);
    FILE *fw = fopen("clone.txt", "w");
    strcpy(clone,strrev(str));
    fprintf( fw,"%s", clone);
    fclose(fw);
    return 0;
}
```



After executing the code



5(b) or:

Solved By EzazC223009-

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

```
int main()
```

```
{
```

```
    FILE *fp=fopen("test.txt","w");
```

```
    int i,n,a[10];
```

```
    printf("Enter 10 Numbers => ");
```

```
    for(i=0; i<10; i++)
```

```
    {
```

```
        scanf("%d",&n);
```

```
        fprintf(fp,"%d ",n);
```

```
    }
```

```
    fclose(fp);
```

```
    printf("\nNumbers in reverse order => ");
```

```
    FILE *fpr=fopen("test.txt","r");
```

```
    for(i=0; i<10; i++)
```

```
    {
```

```
        fscanf(fpr,"%d ", &a[i]);
```

```
    }
```

```
    for(i=9;i>=0;i--)
```

```
    {
```

```
        printf("%d ",a[i]);
```

```
    }
```

```
    printf("\n");
```

```
    fclose(fpr);
```

```
}
```

The image shows a C program in a text editor and its execution in a console window. The program reads 10 numbers into an array and prints them in reverse order. The console output shows the input numbers 1 through 10, followed by the reverse order output 10 down to 1. The process returned 0 and took 5.688 seconds to execute.

```
*Ezaz C223009.c X EzazC223009.c X
#include<stdio.h>
int main()
{
    FILE *fp=fopen("test.txt","w");
    int i,n,a[10];
    printf("Enter 10 Numbers => ");
    for(i=0; i<10; i++)
    {
        scanf("%d",&n);
        fprintf(fp,"%d ",n);
    }
    fclose(fp);
    printf("\nNumbers in reverse order => ");
    FILE *fpr=fopen("test.txt","r");
    for(i=0; i<10; i++)
    {
        fscanf(fpr,"%d ", &a[i]);
    }
    for(i=9;i>=0;i--)
    {
        printf("%d ",a[i]);
    }
    printf("\n");
    fclose(fpr);
}
```

"C:\Users\Saem\Desktop\Solution Final Exam\EzazC223009.exe"

Enter 10 Numbers => 1 2 3 4 5 6 7 8 9 10

Numbers in reverse order => 10 9 8 7 6 5 4 3 2 1

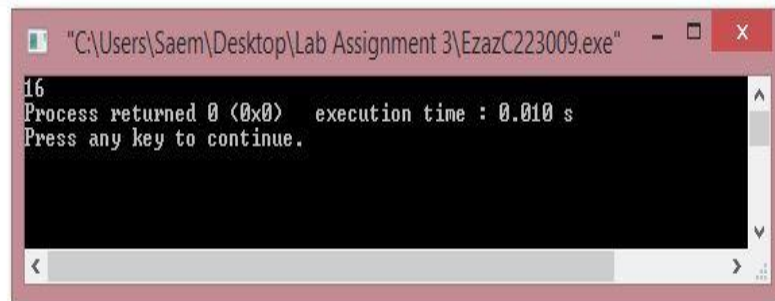
Process returned 0 (0x0) execution time : 5.688 s

Press any key to continue.

5(d) i:

```
#include <stdio.h>
#define MULTI(x,y) x*y
int main()
{
    printf("%d ",MULTI(2+3,3+5));
    return 0;
}
```

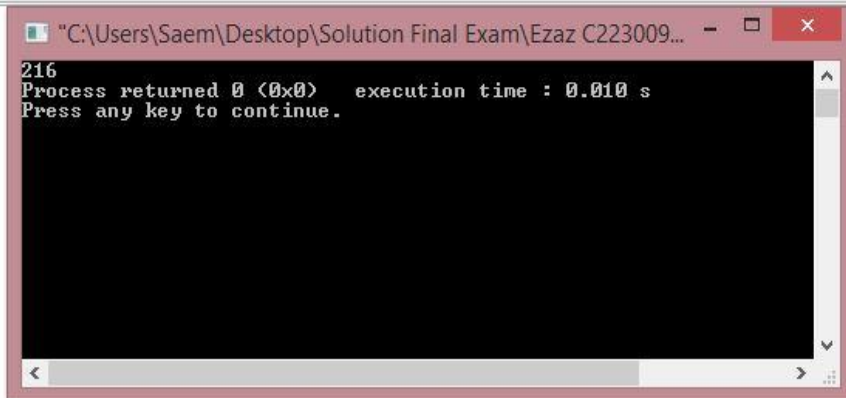
```
#include <stdio.h>
#define MULTI(x,y) x*y
int main()
{
    printf("%d ",MULTI(2+3,3+5));
    return 0;
}
```



5(d) ii:

```
#include<stdio.h>
#define cube(x) x*x*x
int main()
{
    int x=36/cube(6);
    printf("%d",x);
    return 0;
}
```

```
Ezaz C223009.c x
#include<stdio.h>
#define cube(x) x*x*x
int main()
{
    int x=36/cube(6);
    printf("%d",x);
    return 0;
}
```



**Thanks Everyone Assalamualikum**