

**Subject:** PSAP-Problem Solving And Programming

**Name:** Kapil Sangameshwar

**Division:** I

**Roll No:** 32

**PRN:** 12211559

**Assignment No:** 01

**Problem Statement:** Write a program to find the size of different data types using the sizeof operator.

**Theory:** sizeof() is a compile-time operator which can be used to compute the size of its operand. The result of sizeof is of the unsigned integral type which is usually denoted by size\_t. sizeof can be applied to any data type, including primitive types such as integer and floating-point types, pointer types, or compound datatypes such as Structure, union, etc.

**Algorithm:**

1. Declare variables of different data types such as int, float, double, char.
2. Use the sizeof operator to find the size of each variable and store the result in a variable of type size\_t.
3. Use the printf function to print the size of each variable.
4. End the program.

## Program:

```
#include<stdio.h>

int main()
{
    int a;
    float b;
    char c;

    printf("%d\n",sizeof(a));
    printf("%d\n",sizeof(b));
    printf("%d",sizeof(c));
    return 0;
}
```

## Output:

```
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc sizeof.c -o sizeof } ; if ($?) { .\sizeof }
4
4
1
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>
```

**Assignment No:** 02

**Problem Statement:** Write a program to calculate the grade of a student using a logical operator.

**Theory:** We need to use a logical operator to check the range of the marks and determine the corresponding grade. && is and, || is or, used to combine two statements logically.

**Algorithm:**

- 1) Declare a variable to store the marks of the student.
- 2) Read the marks of the student using 'scanf' function.
- 3) Use multiple 'if' statements to check the range of the marks and determine the corresponding grade.
- 4) If marks are within the range specified in the condition, print the corresponding grade.
- 5) If the marks are not within any of the specified ranges, print an error message such as "invalid marks".
- 6) End the program.

## Program:

```
#include <stdio.h>

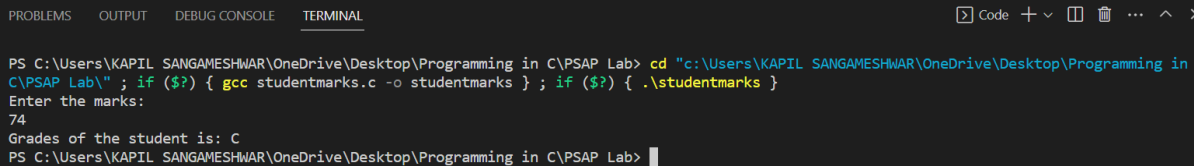
int main() {
    int marks;

    printf("Enter the marks: ");
    scanf("%d", &marks);

    if (marks >= 90) {
        printf("Marks: A\n");
    } else if (marks >= 80) {
        printf("Marks: B\n");
    } else if (marks >= 70) {
        printf("Marks: C\n");
    } else if (marks >= 60) {
        printf("Marks: D\n");
    } else {
        printf("The Marks of the student is: F\n");
    }

    return 0;
}
```

## Output:



```
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc studentmarks.c -o studentmarks } ; if ($?) { .\studentmarks }
Enter the marks:
74
Grades of the student is: C
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>
```

**Assignment No: 03**

**Problem Statement:** Write a program to find out number of Days in a given Month (include Leap Year).

**Theory:** To find out the number of days in a given month, we need to first ask the user to enter the year and month. Next, we need to determine the number of days in the given month. It would be different if the month is February also it would be different for February if it is a leap year. The days would be 30 for months like April, June, September, or November and it would be 31 for the remaining months.

**Algorithm:**

- 1) Declare variable to store the year and month.
- 2) Read the year and month using 'scanf' function.
- 3) Use 'if' statement to check if the month is February or not.
- 4) If it is February, use a nested 'if' statement to check whether it is a Leap year or not, if yes then print 29 days and if not print 28 days.
- 5) For all other month use 'if' statement to check if the month is April, June, September, or November then print 30 days and if not then print 31 days.
- 6) End the program by returning 0 from the 'main' function.

**Program:**

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

int main()
{
    int a, year;

    printf("Enter the year: \n");
    scanf("%d", &year);

    printf("Enter the number of a month: \n");
    scanf("%d", &a);
    if (year % 4 == 0)
    {
        if (a == 1 || a == 3 || a == 5 || a == 7 || a == 8 || a == 10 || a ==
12)
```

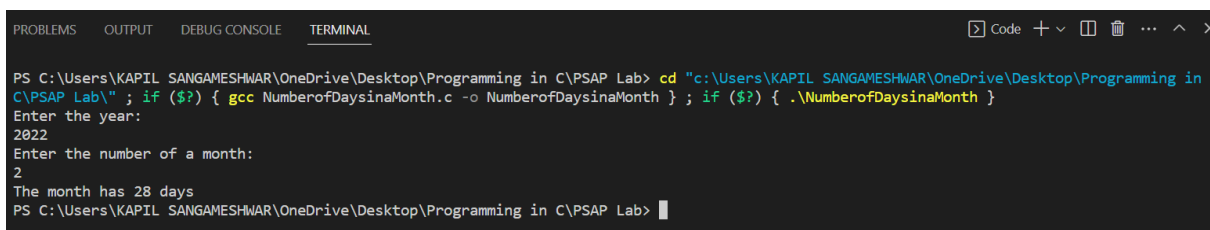
```

    {
        printf("There are 31 days in this month");
    }
    else if (a == 4 || a == 6 || a == 9 || a == 11)
    {
        printf("The month has 30 days");
    }
    else if (a == 2)
    {
        printf("The month has 29 days");
    }
    else
    {
        printf("The number entered is not the number of a month");
    }
}
else
{
    if (a == 1 || a == 3 || a == 5 || a == 7 || a == 8 || a == 10 || a ==
12)
    {
        printf("There are 31 days in this month");
    }
    else if (a == 4 || a == 6 || a == 9 || a == 11)
    {
        printf("The month has 30 days");
    }
    else if (a == 2)
    {
        printf("The month has 28 days");
    }
    else
    {
        printf("The number entered is not the number of a month");
    }
}

return 0;
}

```

## Output:



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc NumberofDaysinaMonth.c -o NumberofDaysinaMonth } ; if ($?) { .\NumberofDaysinaMonth }
Enter the year:
2022
Enter the number of a month:
2
The month has 28 days
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>

```

**Assignment No: 04**

**Problem Statement:** Write a program to find roots of quadratic equation.

**Theory:** A quadratic equation is an equation of the form  $ax^2 + bx + c = 0$ , where  $a$ ,  $b$ , and  $c$  are constants. To find the roots of a quadratic equation, we need to first ask the user to enter the values of  $a$ ,  $b$ ,  $c$ . Next, we need to use the formula for finding the roots of a quadratic equation:  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ . We need to check if the expression under the square root sign is positive or negative. If it is negative, there are no real roots. If it is positive, there are two real roots. We can print both roots.

**Algorithm:**

1. Take input of coefficient variables
2. Calculate the determinant
3. Check the solution type with the given coefficient
4. Calculate the roots accordingly
5. End the program.

**Program:**

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

int main()
{
    float a, b, c, determinant, x1, x2, d, r, i;

    printf("This program solves quadratic equation ax^2+bx+c=0\n");

    printf("Enter value for a\n");
    scanf("%f", &a);

    printf("Enter value for b\n");
    scanf("%f", &b);

    printf("Enter value for c\n");
    scanf("%f", &c);

    determinant = (b * b) - (4 * a * c);
    if (a == 0 && b == 0 && c == 0)
    {
        printf("Infinite solutions");
    }
    else if (a == 0 && b == 0)
```

```
{
    printf("Inconsistent equation");
}
else if (a == 0)
{
    d = -c / b;
    printf("Linear equation with single solution x =%f", d);
}
else if (determinant < 0)
{
    r = -b / (2 * a);
    i = (sqrt(-determinant)) / (2 * a);
    printf("Complex roots :\n");
    printf("x1: %f + %fi\n", r, i);
    printf("x2: %f - %fi", r, i);
}
else if (determinant == 0)
{
    printf("Real and repeated roots :\n");
    x1 = (-b) / (2 * a);
    x2 = (-b) / (2 * a);
    printf("x1: %f\n", x1);
    printf("x2 %f", x2);
}
else
{
    x1 = (-b + sqrt(determinant)) / (2 * a);
    x2 = (-b - sqrt(determinant)) / (2 * a);
    printf("Real and distinct roots :\n");
    printf("x1: %f\n", x1);
    printf("x2 %f", x2);
}
return 0;
}
```



## Output:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Enter value for a
0
Enter value for b
0
Enter value for c
0
Infinite solutions
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc QuadraticEquations.c -o QuadraticEquations } ; if ($?) { .\QuadraticEquations }
This program solves quadratic equation ax^2+bx+c=0
Enter value for a
0
Enter value for b
0
Enter value for c
5
Inconsistent equation
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc QuadraticEquations.c -o QuadraticEquations } ; if ($?) { .\QuadraticEquations }
This program solves quadratic equation ax^2+bx+c=0
Enter value for a
0
Enter value for b
4
Enter value for c
5
Linear equation with single solution x =-1.250000

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
C:\PSAP Lab\" ; if ($?) { gcc QuadraticEquations.c -o QuadraticEquations } ; if ($?) { .\QuadraticEquations }
This program solves quadratic equation ax^2+bx+c=0
Enter value for a
1
Enter value for b
5
Enter value for c
2
Real and distinct roots :
x1: -0.438447
x2: -4.561553
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc QuadraticEquations.c -o QuadraticEquations } ; if ($?) { .\QuadraticEquations }
This program solves quadratic equation ax^2+bx+c=0
Enter value for a
2
Enter value for b
4
Enter value for c
2
Real and repeated roots :
x1: -1.000000
x2: -1.000000
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc QuadraticEquations.c -o QuadraticEquations } ; if ($?) { .\QuadraticEquations }
This program solves quadratic equation ax^2+bx+c=0
Enter value for a
2
Enter value for b
1
Enter value for c
2
Complex roots :
x1: -0.250000 + 0.968246i
x2: -0.250000 - 0.968246i
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>

```

**Assignment No: 05**

**Problem Statement:** Write a program to print a multiplication table of a given number.

**Theory:** To print a multiplication table of a given number, we need to first ask the user for a number for which the multiplication table would be printed. Next, we need to use 'for' loop to iterate through the number from 1 to 10 and then print the result of each multiplication.

**Algorithm:**

- 1) Declare a variable to store the number.
- 2) Read the number using 'scanf' function.
- 3) Use a 'for' loop to iterate through the numbers from 1 to 10.
- 4) Within the 'for' loop, print the result of each multiplication using 'printf' function.
- 5) End the program by returning 0 from the 'main' function.

**Program:**

```
#include <stdio.h>
int main()
{
    int count, n, r;

    printf("Enter a num");
    scanf("%d", &n);
    count = 1;
start:
    if (count <= 10)
    {

        r = n * count;
        printf("%d*%d=%d\n", n, count, r);
        count++;
        goto start;
    }
    return 0;
}
```

## Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Multiplication_Table.c -o Multiplication_Table } ; if ($?) { .\Multiplication_Table }
Enter a num7
7*1=7
7*2=14
7*3=21
7*4=28
7*5=35
7*6=42
7*7=49
7*8=56
7*9=63
7*10=70
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>
```

## Assignment No: 06

**Problem Statement:** Implement a simple calculator in C using switch case construct.

**Theory:** For the implementation of a simple calculator in C, we need to first ask the user to enter numbers with which the operation would be performed also which operation to be done. Next, we need to use a switch case for different operation to be performed.

### Algorithm:

- 1) Declare variable like 'float' and 'char' to store two numbers and an operator.
- 2) Read the user input using 'scanf' function.
- 3) Use switch case statement to handle the different cases for each operator (+, -, \*, /).
- 4) Within each case, perform the appropriate mathematical operation on the two numbers and print the result.
- 5) If the operator input is not one of the four valid operators, print an error message.
- 6) End the program by returning 0 from the 'main' function.

### Program:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int a,b;
    char c;
    printf("Enter two numbers:");
    scanf("%d%d",&a,&b);
    printf("Enter the Operator:");
    scanf(" %c",&c);

    switch(c)
    {
        case '+':
            printf("the sum of the numbers are %d \n",(a+b));
            break;
        case '-':
            printf("the difference of the two numbers is %d \n",(a-b));
            break;

        case '*':
            printf("the multiplication the two numbers is %d \n",(a*b));
            break;
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code + - [ ] [ ] ... - X

PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Switch.c -o Switch } ; if ($?) { .\Switch }
Enter two numbers:5 4
Enter the Operator:+
the sum of the numbers are 9
this is an example of switch case
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Switch.c -o Switch } ; if ($?) { .\Switch }
Enter two numbers:5 4
Enter the Operator:-
the difference of the two numbers is 1
this is an example of switch case
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Switch.c -o Switch } ; if ($?) { .\Switch }
Enter two numbers:5 4
Enter the Operator:*
the multiplication the two numbers is 20
this is an example of switch case
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Switch.c -o Switch } ; if ($?) { .\Switch }
Enter two numbers:5 4
Enter the Operator:/
the division of the two numbers is 1
this is an example of switch case
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Switch.c -o Switch } ; if ($?) { .\Switch }
Enter two numbers:5 4
Enter the Operator:%
the mod of the two numbers is 1
this is an example of switch case
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> |

```

**Assignment No: 07**

**Problem Statement:** Test whether the given number is Armstrong number.

**Theory:** An Armstrong number, also known as a narcissistic number, is a number that is equal to the sum of its own digits each raised to the power of the number of digits.

**Algorithm:**

- 1) Take input of the number.
- 2) Initialize a variable to store the sum of the digits raised to the power of the number of digits.
- 3) Loop through each digit of the number, converting each digit from the string back to an integer and raising it to the power of the number of digits.
- 4) Add each result to the sum.
- 5) Compare the sum to the original number. If they are equal, the number is an Armstrong number. If they are not equal, the number is not an Armstrong number.

**Program:**

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

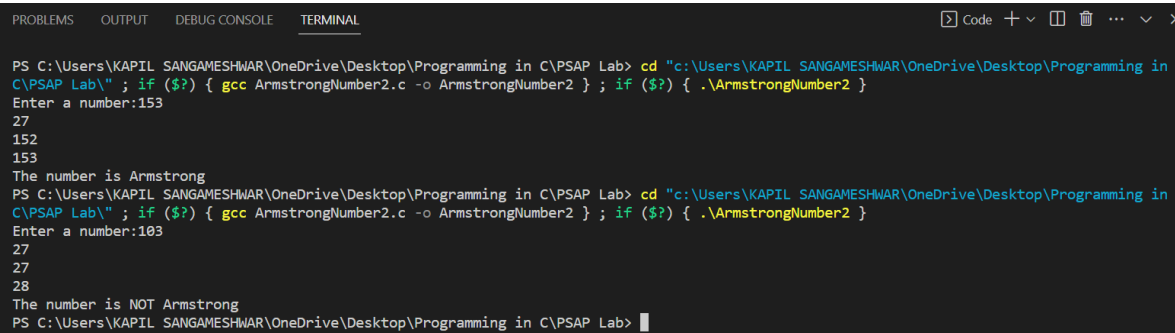
int main()
{
    int a, b, c, d, e, sum = 0;
    printf("Enter a number:");
    scanf("%d", &a);
    c = a;
    d = 0;
    e = a;
    while (e != 0)
    {
        e /= 10;
        d++;
    }

    for (; c != 0; c /= 10)
    {
        b = c % 10;
        sum = sum + round(pow(b, d));
        printf("%d \n", sum);
    }
}
```

```
if (a == sum)
{
    printf("The number is Armstrong");
}
else
{
    printf("The number is NOT Armstrong");
}

return 0;
}
```

## Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc ArmstrongNumber2.c -o ArmstrongNumber2 } ; if ($?) { .\ArmstrongNumber2 }
Enter a number:153
27
152
153
The number is Armstrong
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc ArmstrongNumber2.c -o ArmstrongNumber2 } ; if ($?) { .\ArmstrongNumber2 }
Enter a number:103
27
27
28
The number is NOT Armstrong
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> 
```

**Assignment No:** 08

**Problem Statement:** Implement linear search method using array.

**Theory:** The linear search method is a simple algorithm for finding an element in an array by iterating through the elements one by one until the desired element is found.

**Algorithm:**

- 1) Declare a variable to store ten numbers from the user in the form of array using 'for' loop.
- 2) Read that ten number as an input of the user in 'scanf' function.
- 3) Next, we need the user input for the number they want to search. That search is done by 'for' loop and 'if' statement.
- 4) If the number entered by the user is in the array then print its array location.
- 5) If the entered number is not in the array displayed then print that "No match found".
- 6) End the program by returning 0 to the 'main' function.

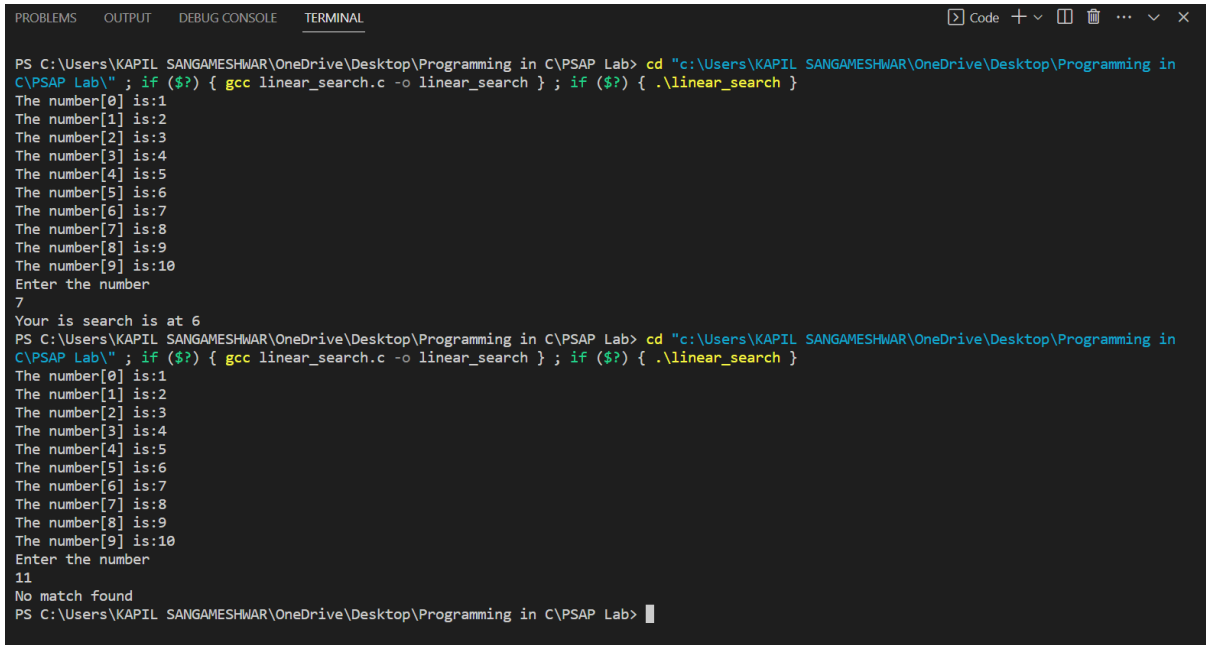
**Program:**

```
#include <stdio.h>
int main()
{
    int number[10];
    int a, i, k=0;
    for (int j = 0; j < 10; j++)
    {
        printf("The number[%d] is:", j);
        scanf("%d", &number[j]);
    }
    printf("Enter the number\n");
    scanf("%d", &a);
    for (i = 0; i < 10; i++)
    {
        if (a == number[i])
        {
            printf("Your is search is at %d\n", i);
            k++;
        }
    }
    if (k==0)
    {
        printf("No match found");
    }
}
```



```
    return 0;  
}
```

## Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc linear_search.c -o linear_search } ; if ($?) { .\linear_search }  
The number[0] is:1  
The number[1] is:2  
The number[2] is:3  
The number[3] is:4  
The number[4] is:5  
The number[5] is:6  
The number[6] is:7  
The number[7] is:8  
The number[8] is:9  
The number[9] is:10  
Enter the number  
7  
Your is search is at 6  
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc linear_search.c -o linear_search } ; if ($?) { .\linear_search }  
The number[0] is:1  
The number[1] is:2  
The number[2] is:3  
The number[3] is:4  
The number[4] is:5  
The number[5] is:6  
The number[6] is:7  
The number[7] is:8  
The number[8] is:9  
The number[9] is:10  
Enter the number  
11  
No match found  
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>
```

## Assignment No: 09

**Problem Statement:** Write a program in C to separate odd and even integers in separate arrays.

**Theory:** In this program first prompts the user to enter the size of the array. It then reads the elements of the array and separates the odd and even numbers into separate arrays using two separate index variables. Finally, it prints the two arrays containing the odd and even numbers.

### Algorithm:

- 1) Declare a variable to store the number in the form of array taken from the user.
- 2) Read the user input in 'scanf' function which will print the array elements using 'for' loop.
- 3) To find Even number in the array elements use 'if' statement as if the array element is divisible by two than print the elements are "Even".
- 4) And if the array elements are not divisible by two than print the elements are "Odd" using 'if' statement and 'for' loop.
- 5) End the program.

### Program:

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <time.h>
#include <stdlib.h>
int main()
{
    int p;

    printf("Enter the number of elements of the array: \n");
    scanf("%d", &p);

    int a[p], even[p], odd[p], i, j = 0, k = 0, n;

    for (i = 0; i < 10; i++)
    {
        printf("Enter the elements of the array: \n");
        scanf("%d", &a[i]);
    }
    for (i = 0; i < 10; i++)
```

```
{
    printf("\n %d", a[i]);
}

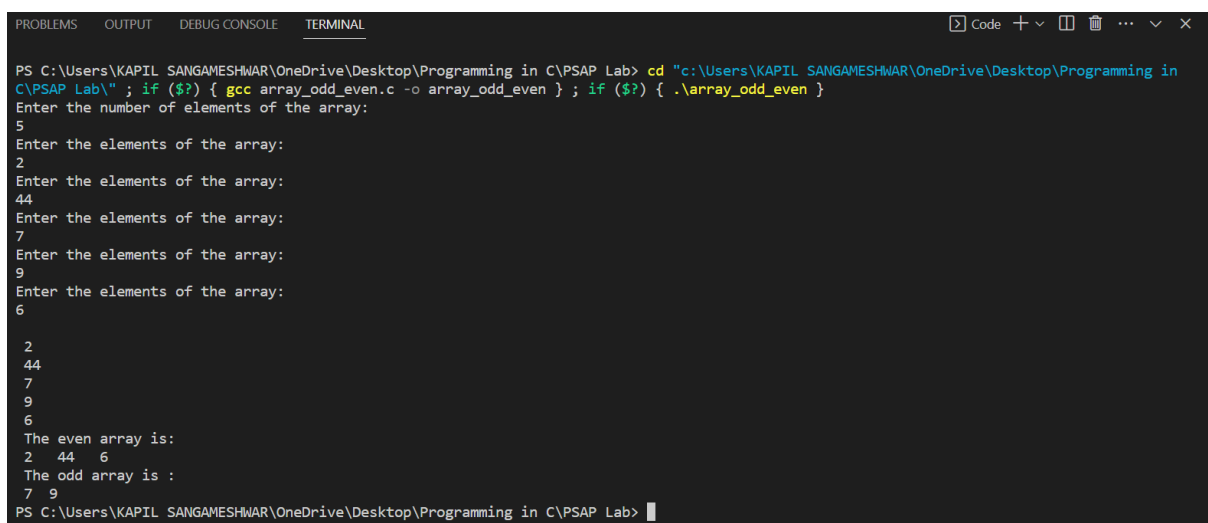
for (i = 0; i < 10; i++)
{
    if (a[i] % 2 == 0)
    {
        even[j] = a[i];
        j++;
    }
    else
    {
        odd[k] = a[i];
        k++;
    }
}

printf("\n The even array is: \n");
for (n = 0; n < j; n++)
{
    printf(" %d ", even[n]);
}

printf("\n The odd array is : \n");
for (n = 0; n < k; n++)
{
    printf(" %d ", odd[n]);
}

return 0;
}
```

## Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab" ; if ($?) { gcc array_odd_even.c -o array_odd_even } ; if ($?) { .\array_odd_even }
Enter the number of elements of the array:
5
Enter the elements of the array:
2
Enter the elements of the array:
44
Enter the elements of the array:
7
Enter the elements of the array:
9
Enter the elements of the array:
6

2
44
7
9
6
The even array is:
2 44 6
The odd array is :
7 9
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> |
```

## Assignment No: 10

**Problem Statement:** Write a program in C to add array elements.

**Theory:** To add the array elements, first we need to consider the array then we have to add them using 'for' loop and a simple statement that "sum = sum + array[]".

### Algorithm:

- 1) Initialize an array 'arr[ ]' to store the elements.
- 2) Read the size of the array 'n' and its elements.
- 3) Initialize a variable 'sum' to store the sum of the array elements and set it to 0.
- 4) Use a for loop to iterate through each element of the array.
- 5) In the loop, add the current element to the 'sum'.
- 6) After the loop, print the 'sum' as the sum of all the elements in the array.
- 7) End the program by returning 0 to the 'main' function.

### Program:

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <time.h>
#include <stdlib.h>
int main()
{
    int n;

    printf("Enter the number of elements in the array: \n");
    scanf("%d",&n);

    int a[10], result = 0, i;

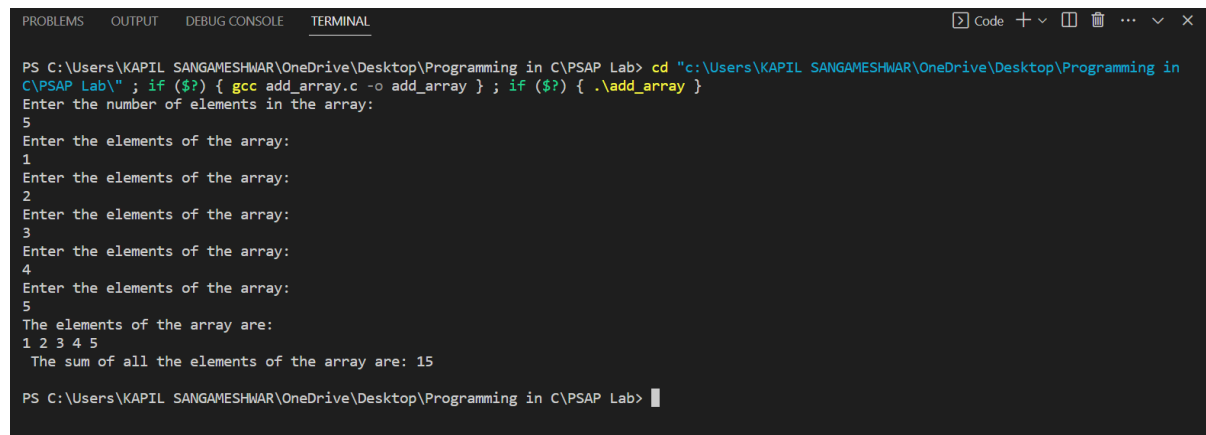
    for (i = 0; i < n; i++)
    {
        printf("Enter the elements of the array: \n");
        scanf("%d", &a[i]);
    }
    printf("The elements of the array are: \n");
    for (i = 0; i < n; i++)
    {
        printf("%d ", a[i]);
    }
}
```

```
}

for (i = 0; i < n; i++)
{
    result += a[i];
}
printf("\n The sum of all the elements of the array are: %d\n ", result);

return 0;
}
```

## Output:



The screenshot shows a terminal window with the following content:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc add_array.c -o add_array } ; if ($?) { .\add_array }
Enter the number of elements in the array:
5
Enter the elements of the array:
1
Enter the elements of the array:
2
Enter the elements of the array:
3
Enter the elements of the array:
4
Enter the elements of the array:
5
The elements of the array are:
1 2 3 4 5
The sum of all the elements of the array are: 15

PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> |
```

## Assignment No:11

**Problem Statement:** Write a program to implement addition, subtraction, multiplication and transpose of matrices.

**Theory:** To write a program to implement addition, subtraction, multiplication and transpose of matrices we need to use switch case to make this all operation in one single code.

### Algorithm:

- 1) Declare two variables to store multidimensional array taken from the user.
- 2) Read the user input by 'scanf' function using 'for' loop for row and columns for both matrices, with taking the values of matrices.
- 3) Display a menu with options for different type of operation like addition, subtraction, multiplication and transpose.
- 4) Based on the user choice, perform the corresponding operations.
- 5) Print the result of matrix based on the user choice.
- 6) End the program.

### Program:

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <time.h>
#include <stdlib.h>

int main()
{
    int n, x1, y1, x2, y2;

    printf("Enter the number of rows: \n");
    scanf("%d", &x1);

    printf("Enter the number of columns: \n");
    scanf("%d", &y1);
    int a[x1][y1];
    printf("Enter the elements of first Matrix: \n");

    for (int i = 0; i < x1; i++)
    {
        for (int j = 0; j < y1; j++)
        {
```

```

        scanf("%d", &a[i][j]);
    }
}
for (int i = 0; i < x1; i++)
{
    for (int j = 0; j < y1; j++)
    {
        printf("%d \t", a[i][j]);
    }
    printf("\n");
}
printf("Enter the number of rows: \n");
scanf("%d", &x2);

printf("Enter the number of columns: \n");
scanf("%d", &y2);
int b[x2][y2];
printf("Enter the elements of second Matrix: \n");

for (int i = 0; i < x2; i++)
{
    for (int j = 0; j < y2; j++)
    {
        scanf("%d", &b[i][j]);
    }
}
for (int i = 0; i < x2; i++)
{
    for (int j = 0; j < y2; j++)
    {
        printf("%d \t", b[i][j]);
    }
    printf("\n");
}
int sum[30][30], sub[30][30], mult[30][30], trans[30][30], trans1[30][30],
w, z;
char i;
printf("Enter the operation you want to use on the given Matrices: \n");
scanf("\n %c", &i);

switch (i)
{
    case '+':
    {
        if (x1 == x1 && y1 == y2)
        {
            for (int m = 0; m < x1; m++)
            {

```

```

        for (int n = 0; n < y1; n++)
        {
            sum[m][n] = a[m][n] + b[m][n];
        }
    }
    for (int m = 0; m < x1; m++)
    {
        for (int n = 0; n < y1; n++)
        {
            printf("%d \t", sum[m][n]);
        }
        printf("\n");
    }
    break;
}

case '-':
{
    if (x1 == x1 && y1 == y2)
    {
        for (int m = 0; m < x1; m++)
        {
            for (int n = 0; n < y1; n++)
            {
                sub[m][n] = a[m][n] - b[m][n];
            }
        }
        for (int m = 0; m < x1; m++)
        {
            for (int n = 0; n < y1; n++)
            {
                printf("%d \t", sub[m][n]);
            }
            printf("\n");
        }
        break;
    }
}

case '*':
{
    if (y1 == x2)
    {
        for (w = 0; w < x1; w++)
        {
            for (z = 0; z < y2; z++)
            {
                int s = 0;

```



```

        for (int l = 0, p = 0; l < y1 && p < x2; l++, p++)
        {
            s += a[w][l] * b[p][z];
        }
        mult[w][z] = s;
    }
}
printf("\n");
for (int o = 0; o < x1; o++)
{
    for (int q = 0; q < y2; q++)
    {
        printf("%d\t", mult[o][q]);
    }
    printf("\n");
}

break;
}
case 't':
{
    if (x1 == x1 && y1 == y2)
    {
        for (int m = 0; m < x1; m++)
        {
            for (int n = 0; n < y1; n++)
            {
                trans[m][n] = a[n][m];
            }
        }
        printf("Transpose of first matrix: \n");
        for (int m = 0; m < x1; m++)
        {
            for (int n = 0; n < y1; n++)
            {
                printf("%d \t", trans[m][n]);
            }
            printf("\n");
        }
    }
    if (x1 == x1 && y1 == y2)
    {
        for (int m = 0; m < x1; m++)
        {
            for (int n = 0; n < y1; n++)
            {
                trans1[m][n] = b[n][m];
            }
        }
    }
}

```

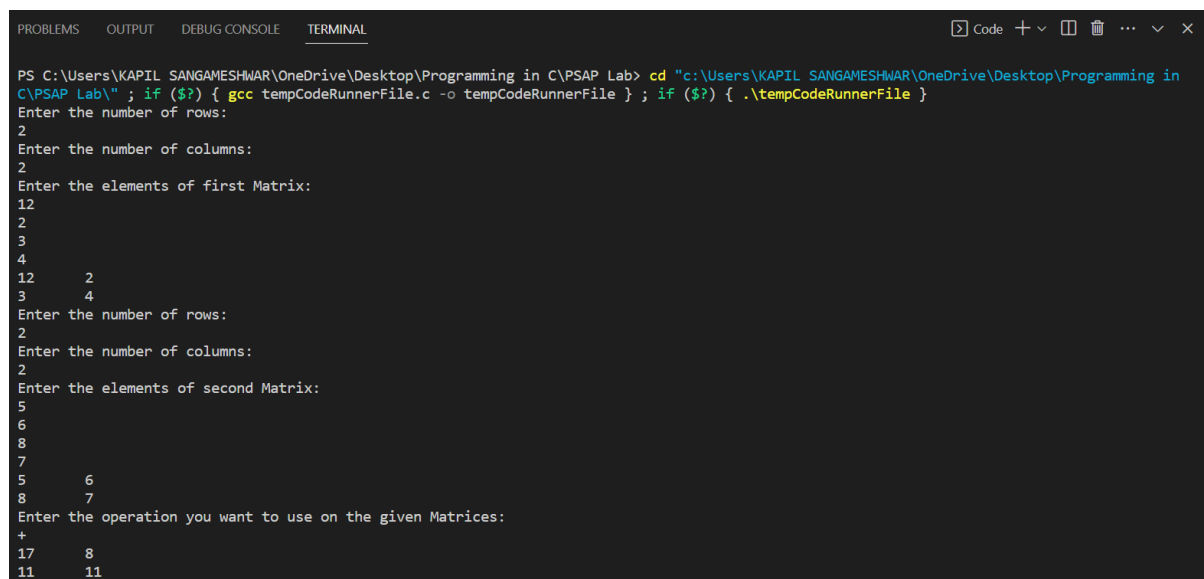
```

    }
}
printf("Transpose of second matrix: \n");
for (int m = 0; m < x1; m++)
{
    for (int n = 0; n < y1; n++)
    {
        printf("%d \t", trans1[m][n]);
    }
    printf("\n");
}
break;
}
}

return 0;
}

```

## Output:



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter the number of rows:
2
Enter the number of columns:
2
Enter the elements of first Matrix:
12
2
3
4
12    2
3     4
Enter the number of rows:
2
Enter the number of columns:
2
Enter the elements of second Matrix:
5
6
8
7
5     6
8     7
Enter the operation you want to use on the given Matrices:
+
17    8
11    11

```

# VISHWAKARMA INSTITUTE OF TECHNOLOGY, PUNE

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter the number of rows:
2
Enter the number of columns:
2
Enter the elements of first Matrix:
5
6
7
8
5      6
7      8
Enter the number of rows:
2
Enter the number of columns:
2
Enter the elements of second Matrix:
5
6
9
8
5      6
9      8
Enter the operation you want to use on the given Matrices:
-
0      0
-2     0
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc two_dimensionalarray_operations.c -o two_dimensionalarray_operations } ; if ($?) { .\two_dimensionalarray_operations }
Enter the number of rows:
2
Enter the number of columns:
2
Enter the elements of first Matrix:
1
2
3
4
1      2
3      4
Enter the number of rows:
2
Enter the number of columns:
2
Enter the elements of second Matrix:
5
6
7
8
5      6
7      8
Enter the operation you want to use on the given Matrices:
*
19      22
43      50
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>
```

# VISHWAKARMA INSTITUTE OF TECHNOLOGY, PUNE

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc two_dimensionalarray_operations.c -o two_dimensionalarray_operations } ; if ($?) { .\two_dimensionalarray_operations }
Enter the number of rows:
2
Enter the number of columns:
2
Enter the elements of first Matrix:
1
2
3
4
1 2
3 4
Enter the number of rows:
2
Enter the number of columns:
2
Enter the elements of second Matrix:
5
6
7
8
5 6
7 8
Enter the operation you want to use on the given Matrices:
t
Transpose of first matrix:
1 3
2 4
Transpose of second matrix:
5 7
6 8
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>
```

## Assignment No: 12

**Problem Statement:** Implement following operations on strings with and without using library functions:

- a. Calculate length of string.
- b. Concatenate two strings.
- c. Compare two strings.
- d. Copy one string into another string.

**Theory:** To write the program which implement the above given operation without using library functions we need to use switch case. Which will help to perform all the four operations in single code.

### Algorithm:

- 1) To calculate the length of a string, first we need to initialize a variable to store the length of the string and set it to 0.
- 2) Use a loop to iterate through each character of the string until a null character is encountered. For each iteration, increment the variable by 1. Then print the length of the string.
- 3) To calculate concatenate of two strings, initialize an array to store the concatenated string.
- 4) Use a loop to copy the characters of the first string in the array initialize. Then copy the characters of the second string into the initialize array, starting from the end of the first string.
- 5) To compare two strings, use a loop to iterate through each character of both strings until a mismatch is found or a null character is encountered in either string.
- 6) For each iteration, compare the character of both strings. If a mismatch is found, return the difference of their values.
- 7) To copy one string into another string, use a loop to iterate through each character of the source string.
- 8) For each iteration, copy the character from the source string into the destination string.
- 9) Next step is to put this all operation in switch case and end the program by returning 0 to the 'main' function.

## Program:

### Case 1: Using Library Functions-

```

#include<stdio.h>
#include<math.h>
#include<conio.h>
#include<time.h>
#include<stdlib.h>
#include<string.h>
int main()
{
    char a[30],b[30],i[30],j;
    printf("Enter the first string: \n");
    scanf("%s",a);

    printf("Enter the second string: \n");
    scanf("%s",b);

    printf("Length of first string is %d \n",strlen(a));
    printf("Length of second string is %d \n",strlen(b));

    printf("Comparison of the size of two strings %d \n",strcmp(a,b));

    printf("The concatenation of the two strings is %s \n",strcat(a,b));

    strcpy(i,a);
    printf("%s",i);

return 0;
}

```

## Output:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc string_functions.c -o string_functions } ; if ($?) { .\string_functions }
Enter the first string:
Kapil
Enter the second string:
Sangameshwar
Length of first string is 5
Length of second string is 12
Comparison of the size of two strings -1
The concatenation of the two strings is KapilSangameshwar
The copied String is: KapilSangameshwar
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>

```

### Case 2: NOT Using Library Functions-

```

#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <time.h>

```

```
#include <stdlib.h>
#include <string.h>
int main()
{
    char a[30], b[30], i, j;

    printf("Enter the First String : \n");
    scanf("%s", a);

    printf("Enter the Second String: \n", b);
    scanf("%s", b);
    /*String length*/
    for (i = 0; a[i] != '\0';)
    {
        i++;
    }
    printf("length of first string: %d \n", i);

    for (j = 0; b[j] != '\0';)
    {
        j++;
    }
    printf("length of second string: %d \n", j);
    int m = i, n = j;
    /*String Copy*/
    char c[30], d[30];
    int k;

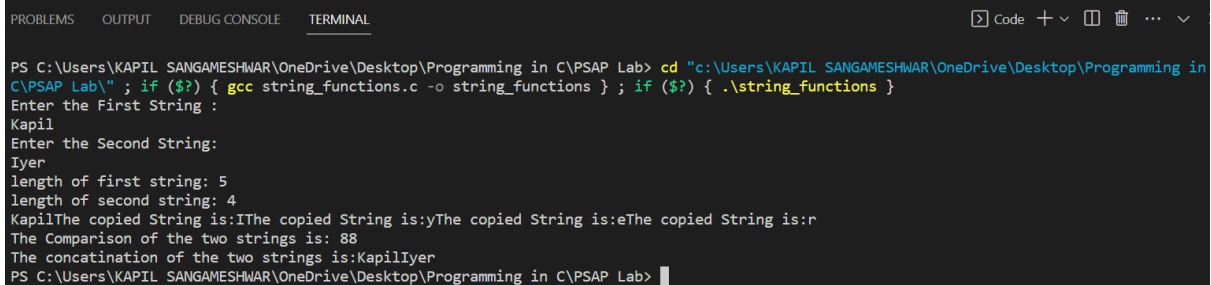
    for (k = 0; k < m; k++)
    {
        c[k] = a[k];

        printf("%c", c[k]);
    }
    for (k = 0; k < n; k++)
    {
        d[k] = b[k];
        printf("%c", d[k]);
    }
    printf("\n");

    /*string compare*/
    int sum = 0;
    for (i = 0; a[i] != '\0'; i++)
    {
        if (a[i] != b[i])
        {
            sum += a[i] - b[i];
        }
    }
}
```

```
    }  
}  
printf("The Comparison of the two strings is: %d \n", sum);  
/*String Concatenate*/  
for (i = 0; i < n; i++)  
{  
    c[m + i] = d[i];  
}  
c[m + n] = '\\0';  
printf("%s", c);  
  
return 0;  
}
```

## Output:



```
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc string_functions.c -o string_functions } ; if ($?) { .\string_functions }  
Enter the First String :  
Kapil  
Enter the Second String:  
Iyer  
length of first string: 5  
length of second string: 4  
KapilThe copied String is:IThe copied String is:yThe copied String is:eThe copied String is:r  
The Comparison of the two strings is: 88  
The concatenation of the two strings is:KapilIyer  
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> █
```



## Assignment No: 13

**Problem Statement:** Write a program in C for binary search using recursion.

**Theory:** Binary search is an efficient algorithm for finding an item in a sorted list of items. It works by dividing the list in half, searching in the half that the item is most likely to be in, and repeating the process until the item is found or it is determined that the item is not in the list.

### Algorithm:

1. Declare a search function
2. Take user input of an sorted array
3. Enter the element to be searched, x
4. In search function, take parameters of array, x, highest and lowest value of array
5. If middle value isn't the element then identify if its higher or lower than x.
6. Then use recursion of search. If middle value is greater, then take the value before middle one as highest value. If it is lower, take next value of middle value as lowest
7. Print the ouput
8. End the program

### Program:

```
#include <stdio.h>

int bsearch(int *array, int key, int low, int high)
{
    int mid = low + (high - low) / 2;
    printf("%d %d %d \n", low, mid, high);
    if (array[mid] == key)
    {
        return mid;
    }
    else if (array[mid] < key)
    {
        bsearch(array, key, mid + 1, high);
    }
    else if (array[mid] > key)
    {

```

```
        bsearch(array, key, low, mid - 1);
    }
    else
    {
        return -1;
    }
}
int main()
{
    int n, i, key, index;
    printf("Enter the number of elements of the array: \n");
    scanf("%d", &n);
    int array[n];

    printf("Enter the elements of the array: \n");

    for (i = 0; i < n; i++)
    {
        scanf("%d", &array[i]);
    }

    for (i = 0; i < n; i++)
    {
        printf(" %d \n", array[i]);
    }
    printf("Enter the key to search: \n");
    scanf("%d", &key);

    index = bsearch(array, key, 0, n - 1);

    if (index >= 0)
    {
        printf("Found the key(%d) at: %d \n", key, index);
    }
    else
    {
        printf("The key not found.");
    }

    return 0;
}
```

## Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  Code + - [ ] [ ] ... ^ X

PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc binary_search.c -o binary_search } ; if ($?) { .\binary_search }
Enter the number of elements of the array:
5
Enter the elements of the array in increasing order:
1
2
3
4
5
1
2
3
4
5
Enter the key to search:
4
0 2 4
3 3 4
Found the key(4) at: 3
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> |
```

## Assignment No: 14

**Problem Statement:** Write a program to implement Fibonacci Series using call by reference method.

**Theory:** The Fibonacci series is a series of numbers in which each number is the sum of the two preceding ones, starting from 0 and 1. In this example, we will use the “call by reference” method to pass the values of the variables between function calls.

### Algorithm:

- 1) Start by declaring a function called ‘fibonacci’ that takes three parameters: two pointers to integers ‘a’ and ‘b’, and an integer ‘n’.
- 2) In the ‘fibonacci’ function, declare a variable ‘c’ to store the sum of ‘a’ and ‘b’, and an integer ‘l’ to keep track of the number of terms generated.
- 3) Use a for loop to generate the next ‘n’ terms of the series. In each iteration of the loop, calculate the next term ‘c’ as the sum of ‘a’ and ‘b’. Update the values of ‘a’ and ‘b’ with ‘b’ and ‘c’, respectively.
- 4) Print the value of ‘c’ at the end of each iteration.
- 5) In the ‘main’ function, prompt the user to enter the number of terms in the series and store it in a variable ‘n’.
- 6) Initialize the first two terms of the series as ‘0’ and ‘1’, and print them to the console.
- 7) Call the fibonacci function, passing ‘a’ and ‘b’ as addresses and ‘n-2’ as the value of ‘n’.
- 8) Print a newline character to end the output.
- 9) End the program.

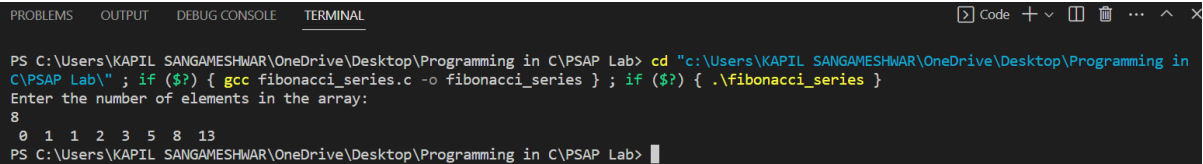
### Program:

```
#include<stdio.h>
int fibonacci(int *array,int n)
{
    for(int i=3;i<n;i++)
    {
        array[i]=array[i-1]+array[i-2];
    }
}

int main()
```

```
{  
    int n;  
  
    printf("Enter the number of elements in the array: \n");  
    scanf("%d",&n);  
    int a[n];  
  
    a[0]=0;  
    a[1]=1;  
    a[2]=1;  
  
    if(n>3)  
    {  
        fibonacci(a, n);  
    }  
    for(int i=0;i<n;i++)  
    {  
        printf(" %d ",a[i]);  
    }  
    return 0;  
}
```

## Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code + - [ ] [ ] ... ^ X  
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc fibonacci_series.c -o fibonacci_series } ; if ($?) { .\fibonacci_series }  
Enter the number of elements in the array:  
8  
0 1 1 2 3 5 8 13  
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> |
```

## Assignment No: 15

**Problem Statement:** Write a program to implement Factorial of number using call by value method

**Theory:** The factorial of a number is the product of all the positive integers less than or equal to that number. To implement the factorial of a number using the call-by-value method, you can create a function that takes an integer as input and returns the factorial of that number. The function can be implemented using a loop that multiplies the current result by each subsequent positive integer until the specified number is reached.

### Algorithm:

- 1) Start by declaring a function called 'factorial' that takes an integer 'n' as its parameter.
- 2) Declare an integer variable 'result' to store the factorial of 'n'. Initialize 'result' to '1'.
- 3) Use a for loop to iterate from '1' to 'n'. In each iteration of the loop, multiply 'result' by the current value of the loop variable.
- 4) Return 'result' as the factorial of 'n'.
- 5) In the 'main' function, prompt the user to enter a number 'n'.
- 6) Call the 'factorial' function, passing 'n' as an argument, and store the result in a variable.
- 7) Print the result.
- 8) End the program.

### Program:

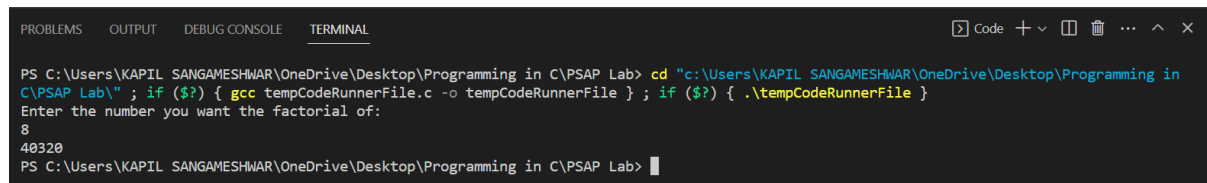
```
#include<stdio.h>

int factorial(int n)
{
    if(n==0 || n==1)
    {
        return 1;
    }
    else{
        return (n*(factorial(n-1)));
    }
}
```

```
int main()
{
    int number;

    printf("Enter the number you want the factorial of: \n");
    scanf("%d",&number);
    factorial(number);
    printf("%d",factorial(number));
}
```

## Output:



The screenshot shows a terminal window with the following content:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  Code  +  -  [ ]  [ ]  ...  ^  X

PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter the number you want the factorial of:
8
40320
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> |
```

## Assignment No: 16

**Problem Statement:** Write a program in C to find the factorial of a given number using pointers.

**Theory :** In this program, we pass a pointer to the 'factorial' function, which calculates the factorial of the number pointed to by the pointer. The pointer is created in the 'main' function by using the address of operator '&' to get the address of the 'number' variable and storing it in 'ptr'. In the 'factorial' function, we dereference the pointer using the indirection operator '\*' to access the value pointed to by the pointer.

### Algorithm:

- 1) Take an integer from the user, which represents the number for which the factorial has to be calculated.
- 2) Declare a pointer 'ptr' that will hold the address of the input number.
- 3) Initialize the pointer 'ptr' with the address of the input number.
- 4) Create a function 'factorial' that takes a pointer as its argument.
- 5) In the 'factorial' function, declare a variable 'result' to store the factorial and initialize it with 1.
- 6) use a for loop to iterate from 1 to the value pointed to by the pointer.
- 7) In each iteration of the loop, multiply the current value of 'result' with the loop counter.
- 8) After the loop, return the value of 'result'.
- 9) Call the 'factorial' function and pass the pointer 'ptr' as an argument.
- 10) Store the result returned by the 'factorial' function in a variable.
- 11) Print the factorial of the input number using the result.

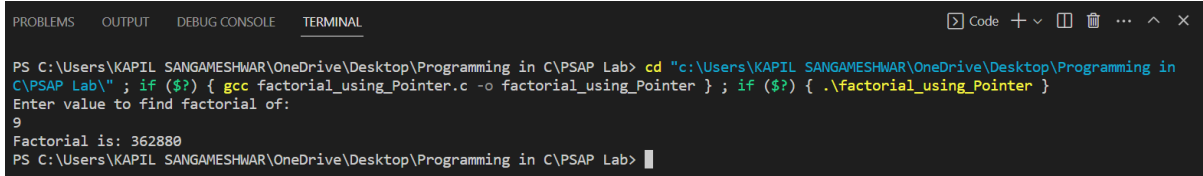
### Program:

```
#include <stdio.h>
int main()
{
    int x,answer=1;
    int i=1;
    int *ptr,*ptrans;
    printf("Enter value to find factorial of: \n");
    scanf("%d",&x);
    ptrans=&answer;
    ptr=&x;
    for(i=1;i<=*ptr;i++)
```



```
{  
    *ptrans=(*ptrans)*i;  
}  
printf("Factorial is: %d",answer);  
return 0;  
}
```

## Output:



The screenshot shows a Visual Studio Code interface with the 'TERMINAL' tab active. The terminal displays the following commands and output:

```
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc factorial_using_Pointer.c -o factorial_using_Pointer } ; if ($?) { .\factorial_using_Pointer }  
Enter value to find factorial of:  
9  
Factorial is: 362880  
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> |
```

## Assignment No: 17

**Problem Statement:** Write a program in C to sort an array using Pointer.

**Theory:** Sorting an array in ascending or descending order is a common task in computer programming. To sort an array using pointers, you can write a function that takes an array and its size as inputs, and sorts the array by swapping its elements as necessary.

### Algorithm:

- 1) In the main function, declare an array with a fixed size.
- 2) Prompt the user to enter the number of elements in the array and store it in a variable
- 3) Prompt the user to enter the elements of the array and store them in the array.
- 4) Use pointers to access the elements of the array and compare and swap them if necessary.
- 5) In the main function, print the sorted array.
- 6) End the program.

### Program:

```
#include <stdio.h>
int main()
{
    int n;
    printf("Enter the size of the array: \n");
    scanf("%d", &n);

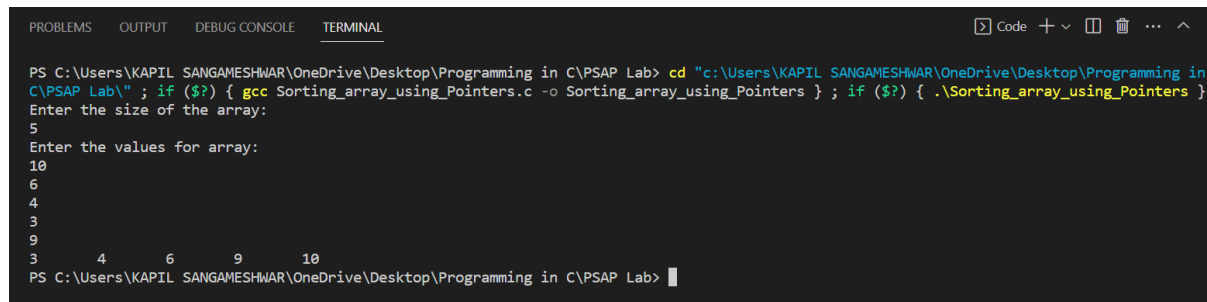
    int a[n], *ptr1, *ptr2, i, temp, j;

    printf("Enter the values for array:");

    for (i = 0; i < 5; i++)
    {
        scanf("%d", &a[i]);
    }
    for (j = 0; j < 4; j++)
    {
        for (i = 0; i <= 4; i++)
        {
            ptr1 = &a[i];
```

```
        ptr2 = &a[i + 1];
        if (*ptr1 > *ptr2)
        {
            temp = *ptr1;
            *ptr1 = *ptr2;
            *ptr2 = temp;
        }
    }
    for (i = 0; i <= 4; i++)
    {
        printf("%d\t", a[i]);
    }
    return 0;
}
```

## Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Sorting_array_using_Pointers.c -o Sorting_array_using_Pointers } ; if ($?) { .\Sorting_array_using_Pointers }
Enter the size of the array:
5
Enter the values for array:
10
6
4
3
9
3      4      6      9      10
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>
```

## Assignment No: 18

**Problem Statement:** Write a C program to count the number of vowels and consonants in a given string using pointers.

**Theory:** String can be sorted in vowels and constants by reading every character and then sorting it accordingly using logical operators.

### Algorithm:

1. Take input from user for a string
2. Find length of the string
3. Loop every element of the string to check with multiple logical operators containing vowel characters
4. Increment a counter variable
5. Print the counter variable for no of vowels and length-counter variable for no of consonants
6. End the program

### Program:

```
#include <stdio.h>
#include <string.h>
int main()
{
    char a[10], *ptr;
    int i, b = 0, length;

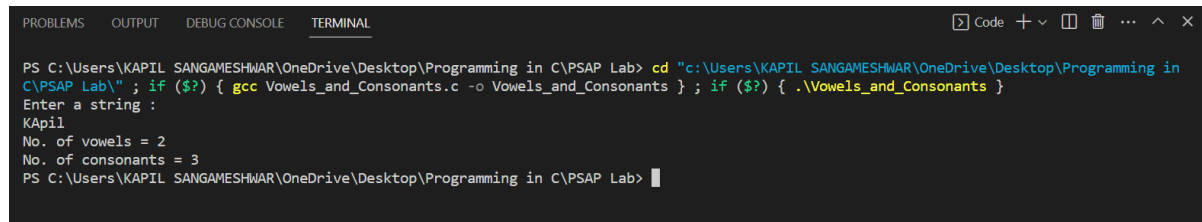
    printf("Enter a string : \n");
    scanf("%s", a);

    length = strlen(a);

    for (i = 0; i <= 10; i++)
    {
        ptr = &a[i];
        if ((*ptr == 'a') || (*ptr == 'A') || (*ptr == 'e') || (*ptr == 'E')
|| (*ptr == 'i') || (*ptr == 'I') || (*ptr == 'o') || (*ptr == 'O') || (*ptr
== 'u') || (*ptr == 'U'))
        {
            b++;
        }
    }
}
```

```
}  
printf("No. of vowels = %d\n", b);  
printf("No. of consonants = %d", length - b);  
return 0;  
}
```

## Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Vowels_and_Consonants.c -o Vowels_and_Consonants } ; if ($?) { .\Vowels_and_Consonants }  
Enter a string :  
KApil  
No. of vowels = 2  
No. of consonants = 3  
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> |
```

## Assignment No: 19

**Problem Statement:** Write a C program to perform basic arithmetic operations using a function pointer.

**Theory:** Pointers access the value stored at a address. To perform arithmetic operations, accessed values can be called and stored in a variable after calculation.

### Algorithm:

1. Take input of two numbers
2. Access the values using pointers and then store their +,-,\*,/,% in different variables
3. Print the calculated values
4. End the program.

### Program:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a, b, *ptr1, *ptr2;
    float c,d;

    printf("Enter two numbers: \n");
    scanf("%d %d", &a, &b);

    ptr1 = &a;
    ptr2 = &b;
    c=(*ptr1)*1.0 / (*ptr2);
    d=(*ptr1) % (*ptr2);

    printf("The sum of the numbers are %d \n", (*ptr1 + *ptr2));

    printf("The difference of the two numbers is %d \n", (*ptr1 - *ptr2));

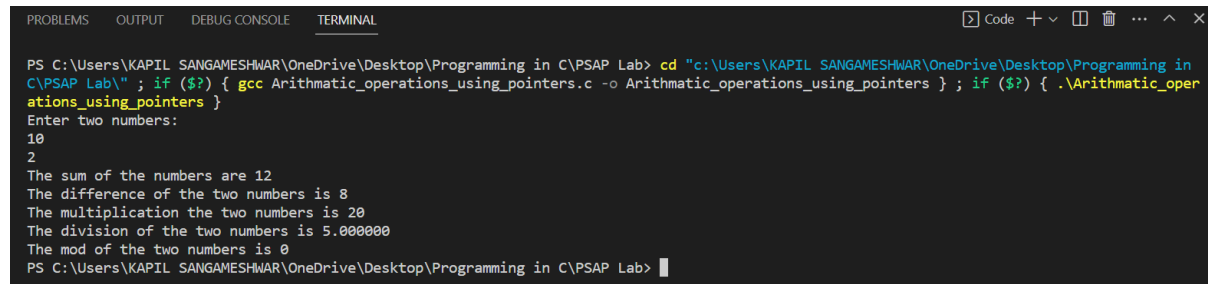
    printf("The multiplication the two numbers is %d \n", ((*ptr1) *
(*ptr2)));

    printf("The division of the two numbers is %f \n", c);
```

```
printf("The mod of the two numbers is %d \n", d);

return 0;
}
```

## Output:



```
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Arithmetic_operations_using_pointers.c -o Arithmetic_operations_using_pointers } ; if ($?) { .\Arithmetic_operations_using_pointers }
Enter two numbers:
10
2
The sum of the numbers are 12
The difference of the two numbers is 8
The multiplication the two numbers is 20
The division of the two numbers is 5.000000
The mod of the two numbers is 0
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> |
```

## Assignment No: 20

**Problem Statement:** Write a C program to swap elements using a double pointer.

**Theory:** Values stored in a variable can be accessed in separate function using pointers using method called call by reference. Then the numbers can be swapped.

### Algorithm:

1. Take input from the user of two numbers
2. Access the values in a declared function then swap them
3. Print the values using pointers
4. End the program

### Program:

```
#include <stdio.h>

void swap(int **a, int **b)
{
    int temp = **a;
    **a = **b;
    **b = temp;
}

int main()
{
    int a, b;

    printf("Enter two numbers: \n");
    scanf("%d %d", &a, &b);

    int *ptr1 = &a, *ptr2 = &b;
    printf("Before swap: a = %d, b = %d\n", *ptr1, *ptr2);
    swap(&ptr1, &ptr2);
    printf("After swap: a = %d, b = %d\n", *ptr1, *ptr2);
    return 0;
}
```

### Output:



# VISHWAKARMA INSTITUTE OF TECHNOLOGY, PUNE

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  Code + - [ ] [X] ... ^ X

PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Swaping_using_pointers.c -o Swaping_using_pointers } ; if ($?) { .\Swaping_using_pointers }
Enter two numbers:
2
9
Before swap: a = 2, b = 9
After swap: a = 9, b = 2
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> |
```

```
#include <stdio.h>

struct data
{
    char Name[50];
    int Roll_No;
    float Marks;
};

int main()
{
    int n;

    struct data information;

    FILE *fptr;

    fptr = (fopen("c:\\Users\\KAPIL
SANGAMESHWAR\\OneDrive\\Desktop\\Programming in C\\PSAP
Lab\\student_info.txt", "w+"));
    printf("\t\t\t\t\t\t\t");
    printf("Enter data of the students\n");
    printf("Enter number of the students: \n");
    scanf("%d", &n);

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

```
{
    printf("Enter name: ");
    scanf("%s", information.Name);
    printf("Enter roll: ");
    scanf("%d", &information.Roll_No);
    printf("Enter marks: ");
    scanf("%f", &information.Marks);
    fprintf(fp, "Student no %d has name %s, roll %d and marks %f \n", i,
information.Name, information.Roll_No, information.Marks);
    printf("\n");
}
fclose(fp);

return 0;
}
```

## Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Student_Information.c -o Student_Information } ; if ($?) { .\Student_Information }
Enter data of the students
Enter number of the students:
2
Enter name: Kapil
Enter roll: 32
Enter marks: 98

Enter name: sadf
Enter roll: 6458
Enter marks: 658

PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>
```



student\_info - Notepad

File Edit View

Student no 1 has name Kapil, roll 32 and marks 98.000000  
Student no 2 has name sarga, roll 646 and marks 165645.000000

## Assignment No: 22

**Problem Statement:** Write a C program to read file contents (Student Name, Roll Number, Marks) and display contents.

**Theory:** Data can be read from a file opened in read mode.

### Algorithm:

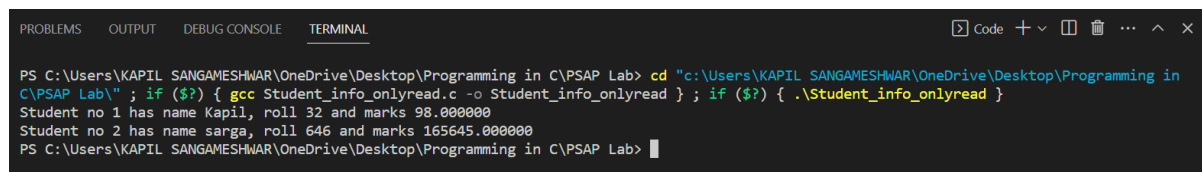
1. Open the file in read mode
2. Print the content till EOF
3. End the program

### Program:

```
#include <stdio.h>

int main()
{
    char Sentence[100];
    FILE *fptr;
    fptr=(fopen("c:\\Users\\KAPIL SANGAMESHWAR\\OneDrive\\Desktop\\Programming in C\\PSAP Lab\\student_info.txt","r"));
    while(fgets(Sentence,sizeof(Sentence),fptr))
    {
        printf("%s",Sentence);
    }
    fgets(Sentence,sizeof(Sentence),fptr);
    return 0;
}
```

### Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab> cd "c:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab\" ; if ($?) { gcc Student_info_onlyread.c -o Student_info_onlyread } ; if ($?) { .\Student_info_onlyread }
Student no 1 has name Kapil, roll 32 and marks 98.000000
Student no 2 has name sarga, roll 646 and marks 165645.000000
PS C:\Users\KAPIL SANGAMESHWAR\OneDrive\Desktop\Programming in C\PSAP Lab>
```