

1. Write a C program to check whether a number is palindrome or not:-

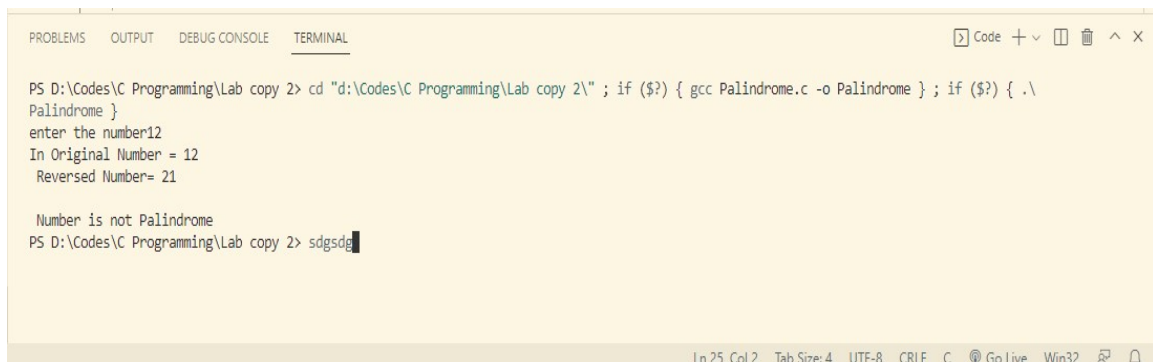
Code:-

```
#include<stdio.h>

int main()
{
    int num, rem, reverse=0, original;
    printf("enter the number");
    scanf("%d",&num);
    printf("In Original Number = %d", num);
    original=num;
    while(num!=0)
    {
        rem = num%10;
        reverse=10*reverse + rem;
        num = num/10;
    }

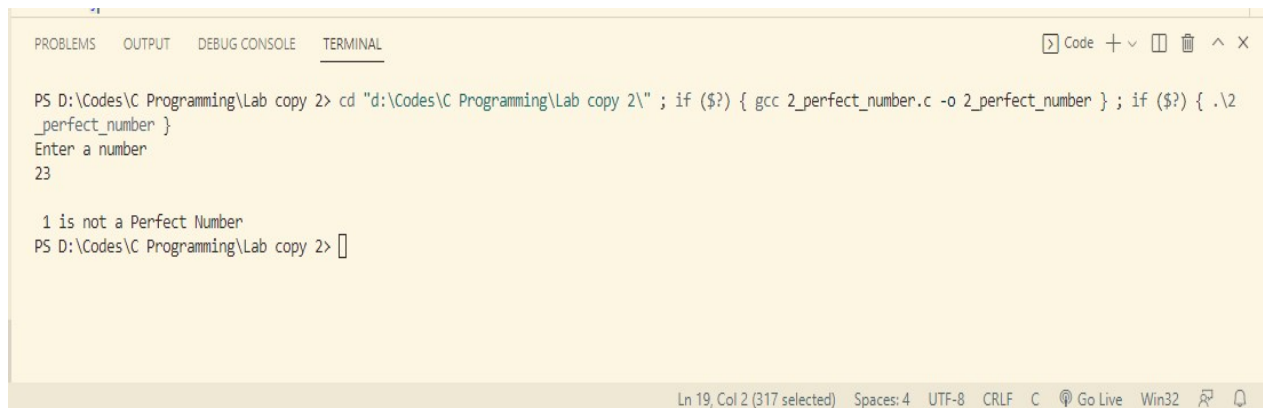
    printf("\n Reversed Number= %d",reverse);
    if(original==reverse)
    {
        printf("\n\n Number is Palindrome");
    }
    else
    {
        printf("\n\n Number is not Palindrome");
    }
}
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc Palindrome.c -o Palindrome } ; if ($?) { .\
Palindrome }
enter the number:12
In Original Number = 12
Reversed Number= 21

Number is not Palindrome
PS D:\Codes\C Programming\Lab copy 2> sdgsgd
```

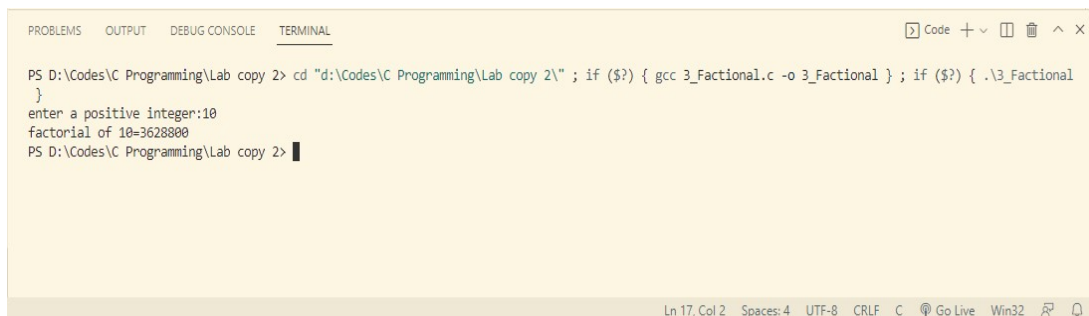


3. Write a program to calculate the factorial of number using recursion:-

Code:-

```
#include<stdio.h>
long int multiplynumbers(int n);
int main()
{
    int n;
    printf("enter a positive integer:");
    scanf("%d",&n);
    printf("factorial of %d=%d",n,multiplynumbers(n));
    return 0;
}
long int multiplynumbers(int n)
{
    if(n>=1)
        return n*multiplynumbers(n-1);
    else
        return 1;
}
```

Output:-



The screenshot shows a terminal window with the following content:

```
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 3_Factional.c -o 3_Factional } ; if ($?) { .\3_Factional
}
enter a positive integer:10
factorial of 10=3628800
PS D:\Codes\C Programming\Lab copy 2> █
```

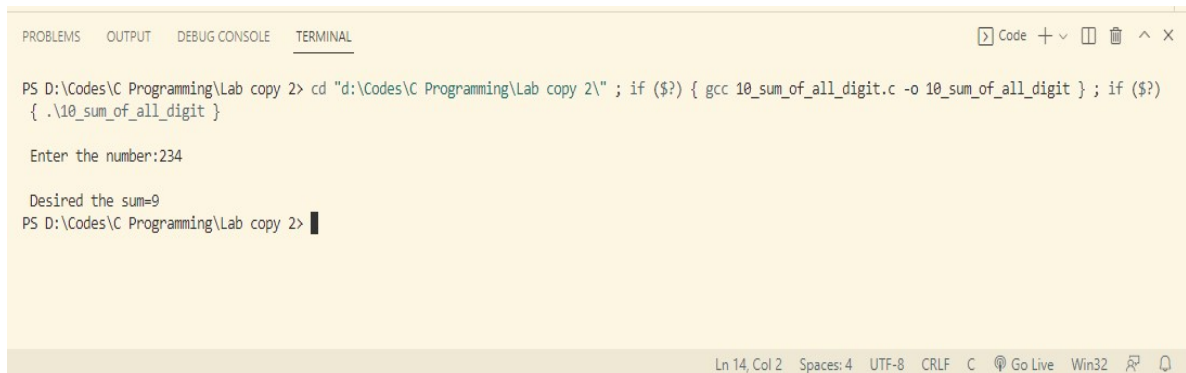
The terminal window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The status bar at the bottom indicates 'Ln 17, Col 2', 'Spaces: 4', 'UTF-8', 'CRLF', 'C', 'Go Live', 'Win32', and icons for search and help.

4. Write a C program to the sum of all the digits of a number:-

Code:-

```
#include<stdio.h>
int main(void)
{
    int number,sum=0,rem;
    printf("\n Enter the number:");
    scanf("%d",&number);
    while(number>0)
    {
        rem=number%10;
        sum=sum+rem;
        number=number/10;
    }
    printf("\n Desired the sum=%d",sum);
}
```

Output:-



The screenshot shows a terminal window with the following content:

```
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 10_sum_of_all_digit.c -o 10_sum_of_all_digit } ; if ($?) { .\10_sum_of_all_digit }
```

Enter the number:234

Desired the sum=9

PS D:\Codes\C Programming\Lab copy 2> █

The terminal window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The status bar at the bottom shows "Ln 14, Col 2", "Spaces: 4", "UTF-8", "CRLF", "C", "Go Live", "Win32", and icons for a refresh and a bell.

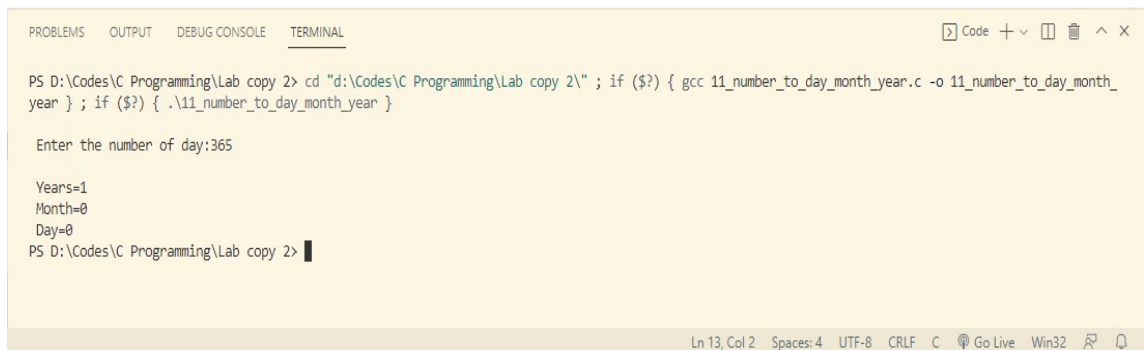
5. Write a C program to convert number of days to Year, Month and Days:-

Code:-

```
#include<stdio.h>

int main(void)
{
    int num,year,mon,day;
    printf("\n Enter the number of day:");
    scanf("%d",&num);
    year=num/365;
    mon=(num-year*365)/30;
    day=(num-year*365-mon*30);
    printf("\n Years=%d",year);
    printf("\n Month=%d",mon);
    printf("\n Day=%d",day);
}
```

Output:-



The screenshot shows a terminal window with a light yellow background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active. The terminal shows the following commands and output:

```
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 11_number_to_day_month_year.c -o 11_number_to_day_month_year } ; if ($?) { .\11_number_to_day_month_year }
```

Enter the number of day:365

Years=1
Month=0
Day=0

PS D:\Codes\C Programming\Lab copy 2> █

At the bottom of the terminal window, there is a status bar showing 'Ln 13, Col 2', 'Spaces: 4', 'UTF-8', 'CRLF', 'C', 'Go Live', 'Win32', and some icons.

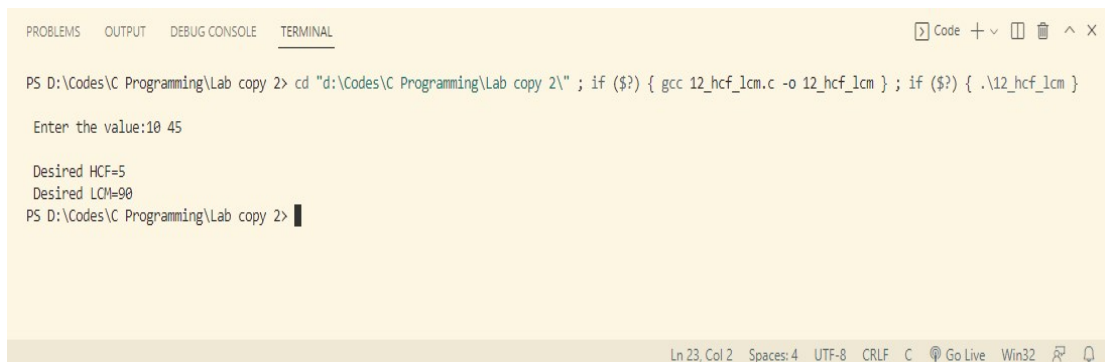
6. Write a C program to calculate HCF & LCM:-

Code:-

```
#include<stdio.h>

int main(void)
{
    int x,y,rem,prod;
    printf("\n Enter the value:");
    scanf("%d %d",&x,&y);
    if(x<y)
    {
        x=x+y;
        y=x-y;
        x=x-y;
    }
    prod=x*y;
    rem=x%y;
    while(rem!=0)
    {
        x=y;
        y=rem;
        rem=x%y;
    }
    printf("\n Desired HCF=%d",y);
    printf("\n Desired LCM=%d",prod/y);
}
```

Output:-



The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 12_hcf_lcm.c -o 12_hcf_lcm } ; if ($?) { .\12_hcf_lcm }

Enter the value:10 45

Desired HCF=5
Desired LCM=90
PS D:\Codes\C Programming\Lab copy 2> |
```

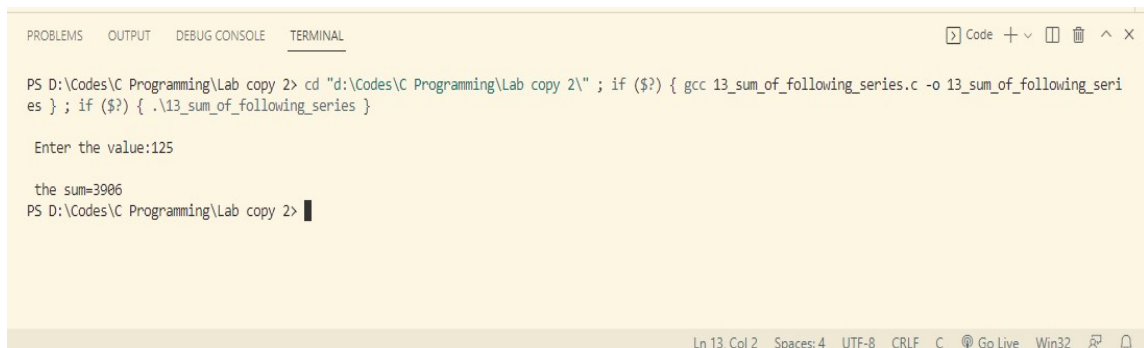
The terminal window has a title bar with 'Code', '+', and icons for window management. The status bar at the bottom shows 'Ln 23, Col 2', 'Spaces: 4', 'UTF-8', 'CRLF', 'C', 'Go Live', 'Win32', and icons for search and refresh.

7. Write a C program to find the sum of the following series: $2+4+6+8+\dots+n$ terms:-

Code:-

```
#include<stdio.h>
int main(void)
{
    int n,i=2,sum=0;
    printf("\n Enter the value:");
    scanf("%d",&n);
    while(i<=n)
    {
        sum=sum+i;
        i=i+2;
    }
    printf("\n the sum=%d",sum);
}
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 13_sum_of_following_series.c -o 13_sum_of_following_series } ; if ($?) { .\13_sum_of_following_series }

Enter the value:125

the sum=3906
PS D:\Codes\C Programming\Lab copy 2> |
```

Ln 13, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32

8. Write a C program to add odd numbers in a given range:-

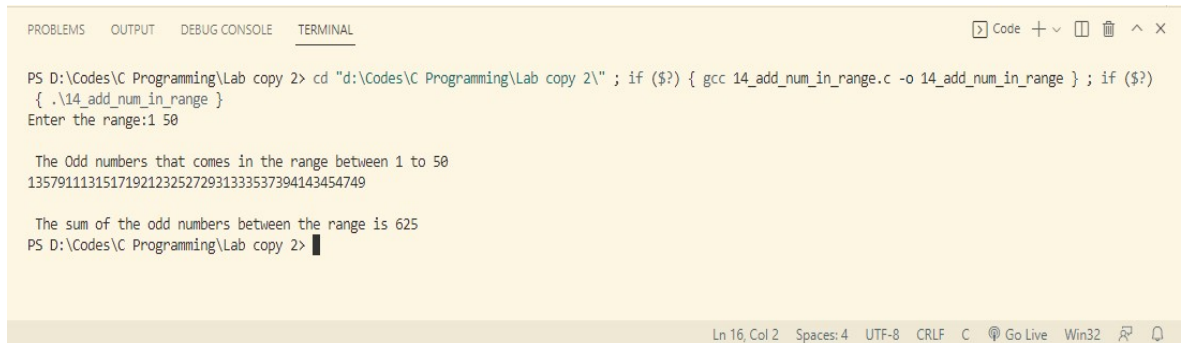
Code:-

```
#include<stdio.h>

int main()
{
    int a,b,i;
    long int sum=0;
    printf("Enter the range:");
    scanf("%d %d",&a,&b);
    printf("\n The Odd numbers that comes in the range
between %d to %d\n",a,b);

    for(i=a;i<=b;i++)
    if(i%2==1)
    {
        sum=sum+i;
        printf("%d",i);
    }
    printf("\n\n The sum of the odd numbers between the
range is %d",sum);
}
```

Output:-

A screenshot of a terminal window showing the execution of a C program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The command prompt shows the user running a gcc command to compile and execute a file named 14_add_num_in_range.c. The program prompts the user to enter a range, and the user enters 1 50. The program then displays the odd numbers in that range and their sum.

```
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 14_add_num_in_range.c -o 14_add_num_in_range } ; if ($?) { .\14_add_num_in_range }
Enter the range:1 50

The Odd numbers that comes in the range between 1 to 50
135791113151719212325272931333537394143454749

The sum of the odd numbers between the range is 625
PS D:\Codes\C Programming\Lab copy 2> |
```


9. Write a c programme to perform Peterson number or not:-

Code:-

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

```
int main()
```

```
{
```

```
    int num,temp,rem,sum=0,fact=1;
```

```
    int i;
```

```
    printf("Enter a number :");
```

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

```
    temp=num;
```

```
    while(temp!=0)
```

```
    {
```

```
        rem=temp%10;
```

```
        for(i=1;i<=rem;i++)
```

```
        fact*=i;
```

```
        sum+=fact;
```

```
        fact=1;
```

```
        temp/=10;
```

```
    }
```

```
    if(num==sum)
```

```
        printf("%d is a Peterson Number",num);
```

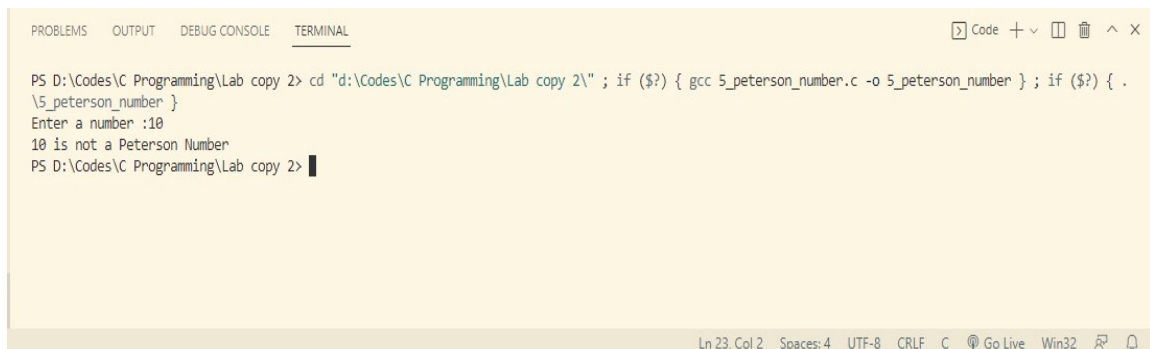
```
    else
```

```
        printf("%d is not a Peterson Number",num);
```

```
    return 0;
```

```
}
```

Output:-



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 5_peterson_number.c -o 5_peterson_number } ; if ($?) { .
5_peterson_number }
Enter a number :10
10 is not a Peterson Number
PS D:\Codes\C Programming\Lab copy 2> █
```

Ln 23, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32 🔍 🔄

10. Write a program to calculate binary to decimal:-

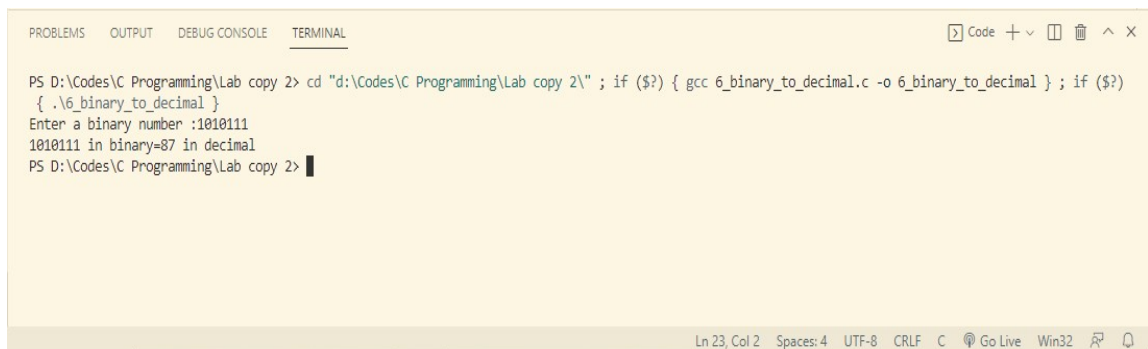
Code:-

```
#include<stdio.h>
#include<math.h>
int convert(long long);
int main()
{
    long long n;
    printf("Enter a binary number :");
    scanf("%d",&n);
    printf("%d in binary=%d in decimal",n,convert(n));
    return 0;
}

int convert(long long n)
{
    int dec=0,i=0,rem;

    while(n!=0)
    {
        rem=n%10;
        n/=10;
        dec+=rem*pow(2,i);
        ++i;
    }
}
```

Output:-



The screenshot shows a terminal window with a yellow background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL', with 'TERMINAL' being the active tab. The terminal displays the following text:

```
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 6_binary_to_decimal.c -o 6_binary_to_decimal } ; if ($?) { .\6_binary_to_decimal }
Enter a binary number :1010111
1010111 in binary=87 in decimal
PS D:\Codes\C Programming\Lab copy 2> █
```

At the bottom of the terminal window, there is a status bar showing 'Ln 23, Col 2', 'Spaces: 4', 'UTF-8', 'CRLF', 'C', 'Go Live', 'Win32', and some icons.

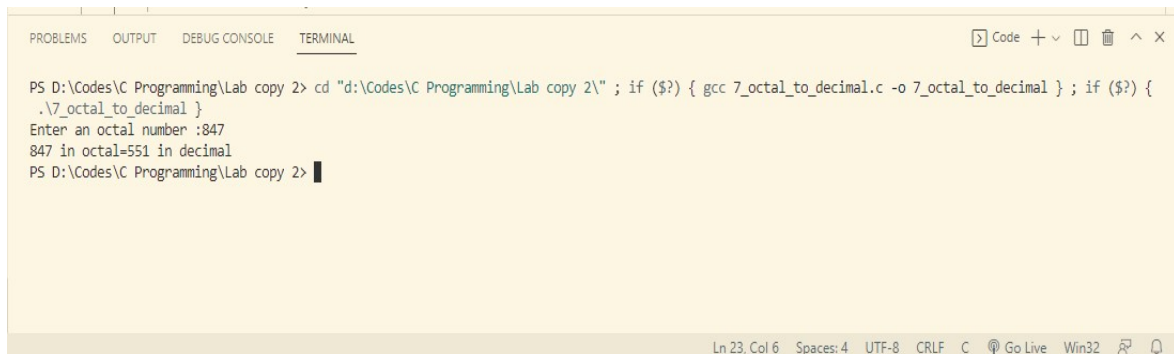
11. Write a program to convert octal to decimal:-

Code:-

```
#include<stdio.h>
#include<math.h>
long long convertOctalToDecimal(int octalNumber);
int main()
{
    int octalNumber;
    printf("Enter an octal number :");
    scanf("%d",&octalNumber);
    printf("%d in octal=%lld in
decimal",octalNumber,convertOctalToDecimal(octalNumber));
    return 0;
}

long long convertOctalToDecimal(int octalNumber)
{
    int decimalNumber=0,i=0;
    while(octalNumber!=0)
    {
        decimalNumber+=(octalNumber%10)*pow(8,i);
        ++i;
        octalNumber/=10;
    }
    i=1;
    return decimalNumber;
}
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 7_octal_to_decimal.c -o 7_octal_to_decimal } ; if ($?) {
.\7_octal_to_decimal }
Enter an octal number :847
847 in octal=551 in decimal
PS D:\Codes\C Programming\Lab copy 2>

Ln 23, Col 6 Spaces: 4 UTF-8 CRLF C Go Live Win32
```

12. Write a c program to reverse a sentence using recursion:-

Code:-

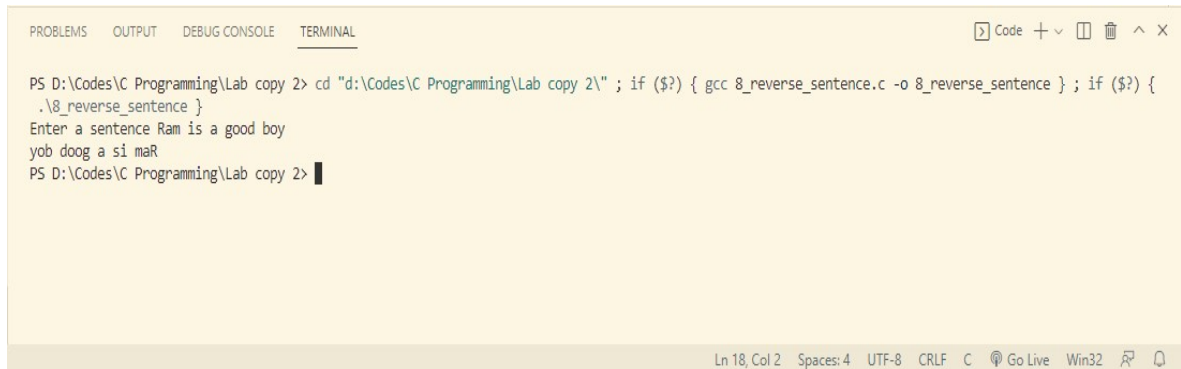
```
#include<stdio.h>

void reverseSentence();

int main()
{
    printf("Enter a sentence");
    reverseSentence();
    return 0;
}

void reverseSentence()
{
    char c;
    scanf("%c",&c);
    if(c!='\n')
    {
        reverseSentence();
        printf("%c",c);
    }
}
```

Output:-



The screenshot shows a terminal window with the following content:

```
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 8_reverse_sentence.c -o 8_reverse_sentence } ; if ($?) {
.\8_reverse_sentence }
Enter a sentence Ram is a good boy
yob doog a si maR
PS D:\Codes\C Programming\Lab copy 2> █
```

The terminal output demonstrates the program's behavior: it prompts the user to enter a sentence, reads the input "Ram is a good boy", and then prints the reversed sentence "yob doog a si maR".

13. Write a c program to swap elements using call by reference:-

Code:-

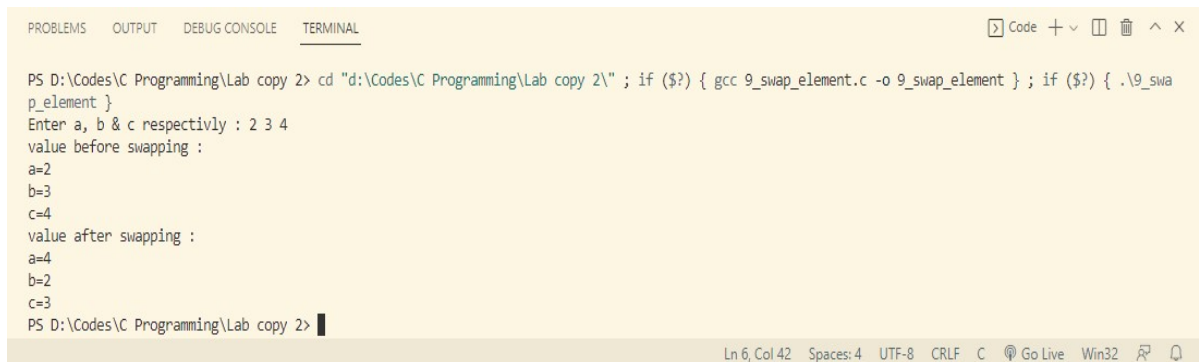
```
#include<stdio.h>

void cyclicswap(int *a,int *b,int *c);

int main()
{
    int a,b,c;
    printf("Enter a, b & c respectively : ");
    scanf("%d%d%d",&a,&b,&c);
    printf("value before swapping : \n");
    printf("a=%d\nb=%d\nc=%d\n",a,b,c);
    cyclicswap(&a,&b,&c);
    printf("value after swapping : \n");
    printf("a=%d\nb=%d\nc=%d\n",a,b,c);
    return 0;
}

void cyclicswap(int *n1,int *n2,int *n3)
{
    int temp;
    temp=*n2;
    *n2=*n1;
    *n1=*n3;
    *n3=temp;
}
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 9_swap_element.c -o 9_swap_element } ; if ($?) { .\9_swap_element }
Enter a, b & c respectively : 2 3 4
value before swapping :
a=2
b=3
c=4
value after swapping :
a=4
b=2
c=3
PS D:\Codes\C Programming\Lab copy 2>
Ln 6, Col 42 Spaces: 4 UTF-8 CRLF C Go Live Win32
```

14. Write a C program to find the number is Armstrong number or not:-

Code:-

```
#include<stdio.h>

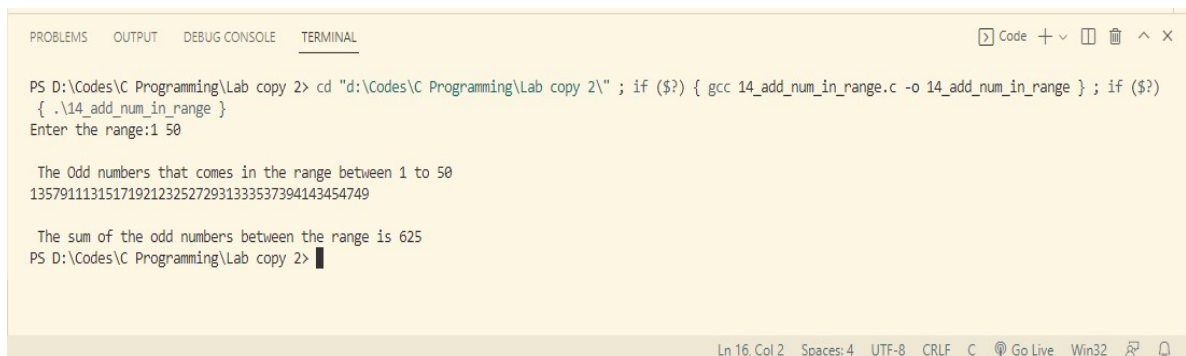
int main()
{
    int number, sum=0, lastDigit, temp;
    printf("Enter a number: ");
    scanf("%d", &number);
    temp = number;

    while (temp!=0)
    {
        lastDigit = temp%10;
        sum = sum + (lastDigit * lastDigit * lastDigit);
        temp = temp/10;
    }

    if (sum == number)
    {
        printf("\n The Armstrong number is = %d", number);
    }
    else
        printf("\n %d is not an Armstrong number\n", number);

    return 0;
}
```

Output:-

A screenshot of a Windows command prompt window showing the execution of a C program. The window title is "C:\Program Files\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\14.29.30133\bin\Hostx64-x64\cmd.exe". The command prompt shows the user entering the range 1 to 50, and the program outputting the list of odd numbers in that range and their sum, 625. The status bar at the bottom indicates the current line and column, as well as the encoding and other settings.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 14_add_num_in_range.c -o 14_add_num_in_range } ; if ($?) { .\14_add_num_in_range }
Enter the range:1 50

The Odd numbers that comes in the range between 1 to 50
135791113151719212325272931333537394143454749

The sum of the odd numbers between the range is 625
PS D:\Codes\C Programming\Lab copy 2>
```

15. Write a C program to take n numbers from user and store them in an array and print the elements:-

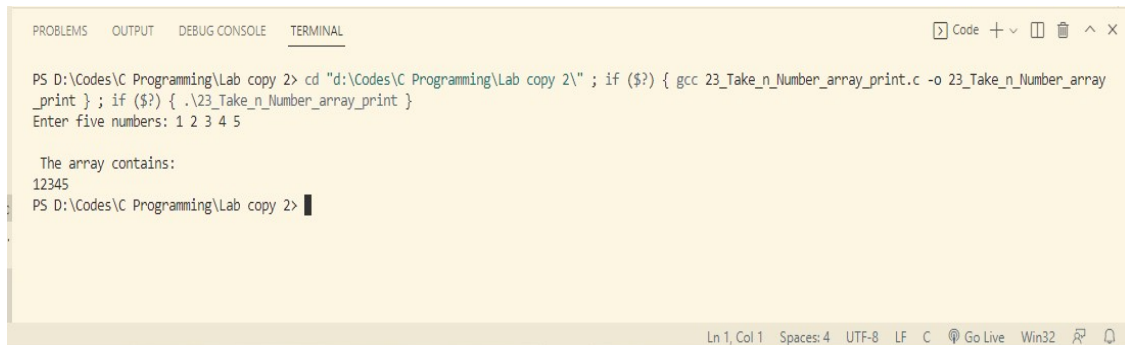
Code:-

```
#include<stdio.h>

int main()
{
    int a[5], i;
    printf("Enter five numbers: ");
    for ( i = 0; i < 5; i++)
    {
        scanf("%d", &a[i]);
    }
    printf("\n The array contains: \n");
    for ( i = 0; i < 5; i++)
    {
        printf("%d", a[i]);
    }

    return 0;
}
```

Output:-



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 23_Take_n_Number_array_print.c -o 23_Take_n_Number_array_print } ; if ($?) { .\23_Take_n_Number_array_print }
Enter five numbers: 1 2 3 4 5

The array contains:
12345
PS D:\Codes\C Programming\Lab copy 2> |
```

Ln 1, Col 1 Spaces: 4 UTF-8 LF C Go Live Win32 🔍 🔔

16. Write a C program to perform matrix multiplication operation using 2d

array:-

Code:-

```
#include<stdio.h>
int main()
{
    int a[10][10], b[10][10], c[10][10], n, i, j, k;
    printf("Enter the value of N (N<=10): ");
    scanf("%d", &n);
    printf("\n Enter the element of Matrix-A: ");
    for ( i = 0; i < n; i++)
    {
        for ( j = 0; j < n; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }

    printf("\n Enter the elements of Matrix-B: ");
    for ( i = 0; i < n; i++)
    {
        for ( j = 0; j < n; j++)
        {
            scanf("%d", &b[i][j]);
        }
    }

    for ( i = 0; i < n; i++)
    {
        for ( j = 0; j < n; j++)
        {
            c[i][j] = 0;
            for ( k = 0; k < n; k++)
            {
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }

    printf("\n The product of two Matrices is: ");
```



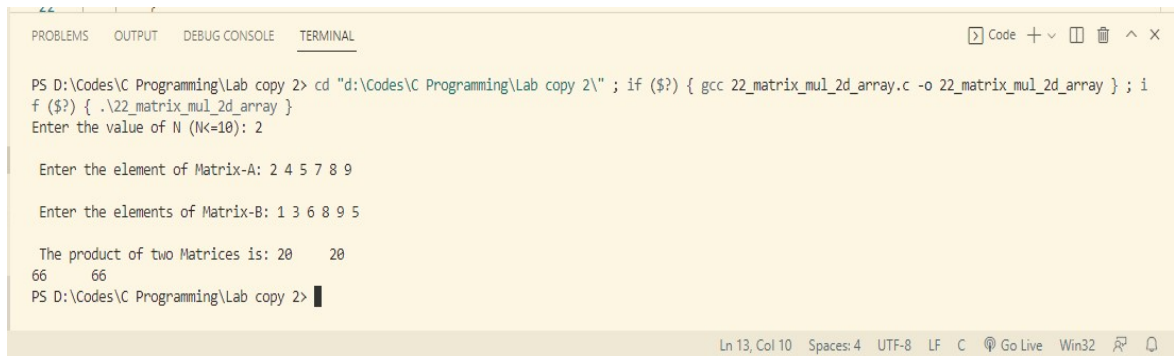
```

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

}

```

Output:-



```

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

PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 22_matrix_mul_2d_array.c -o 22_matrix_mul_2d_array } ; i
f ($?) { .\22_matrix_mul_2d_array }
Enter the value of N (N<=10): 2

Enter the element of Matrix-A: 2 4 5 7 8 9

Enter the elements of Matrix-B: 1 3 6 8 9 5

The product of two Matrices is: 20    20
66    66
PS D:\Codes\C Programming\Lab copy 2>

```

Ln 13, Col 10 Spaces: 4 UTF-8 LF C Go Live Win32

17. Write a program to convert Decimal to Octal:-

Code:-

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

int convertDecimalToOctal(int decimalNumber);

int main()
{
    int decimalNumber;
    printf("Enter the Decimal Number: ");
    scanf("%d", &decimalNumber);
    printf("%d in decimal = %d in octal",
decimalNumber,convertDecimalToOctal(decimalNumber));
    return 0;
}

int convertDecimalToOctal(int decimalNumber)
{
    int octalNumber = 0, i=1;
    while (decimalNumber !=0)
    {
        octalNumber +=(decimalNumber%8)*i;
        decimalNumber /=8;
        i*=10;
    }
    return octalNumber;
}
```

Output:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\"; if ($?) { gcc 21_decimal_to_octal.c -o 21_decimal_to_octal }; if ($?) { .\21_decimal_to_octal }
Enter the Decimal Number: 45
45 in decimal = 55 in octal
PS D:\Codes\C Programming\Lab copy 2> |
```

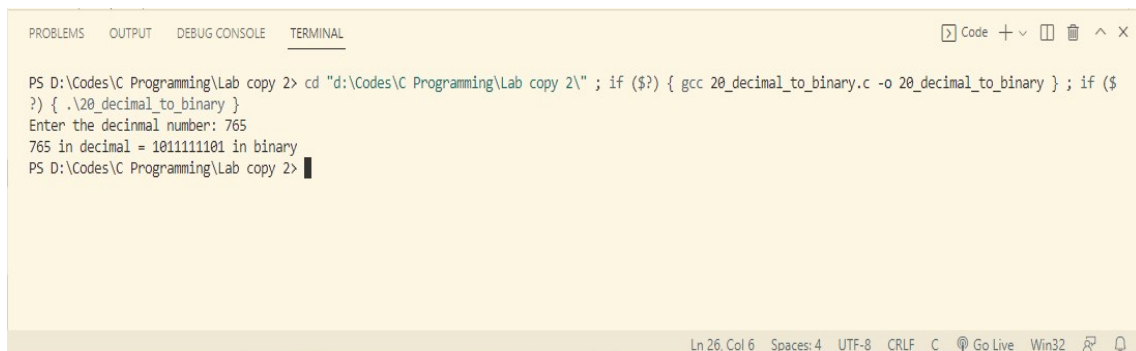
18. Write a program to convert Decimal to Binary:-

Code:-

```
#include<stdio.h>
#include<math.h>
long long convert(int);
int main()
{
    int n, bin;
    printf("Enter the decimal number: ");
    scanf("%d",&n);
    bin = convert(n);
    printf("%d in decimal = %d in binary", n, bin);
    return 0;
}

long long convert(int n)
{
    long long bin = 0;
    int rem, i=1;
    while (n!=0)
    {
        rem = n%2;
        n/=2;
        bin+= rem * i;
        i*=10;
    }
    return bin;
}
```

Output:-



The screenshot shows a terminal window with a yellow background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL', with 'TERMINAL' being the active tab. The terminal displays the following text:

```
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 20_decimal_to_binary.c -o 20_decimal_to_binary } ; if ($?) { .\20_decimal_to_binary }
Enter the decimal number: 765
765 in decimal = 1011111101 in binary
PS D:\Codes\C Programming\Lab copy 2> █
```

At the bottom of the terminal window, there is a status bar showing 'Ln 26, Col 6', 'Spaces: 4', 'UTF-8', 'CRLF', 'C', 'Go Live', 'Win32', and some icons.

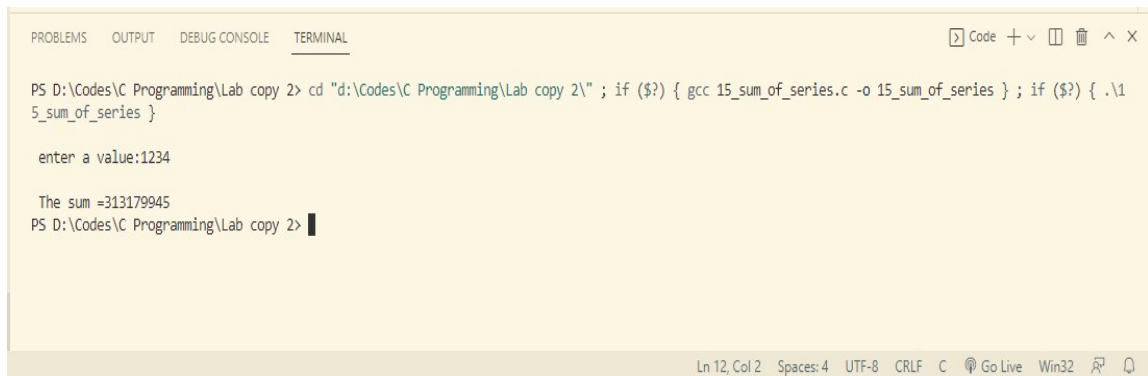
19. Write a program in C to find out the sum of following series

12+32+52+72+.....n terms:-

Code:-

```
#include<stdio.h>
#include<math.h>
int main(void){
    int n, i=1, sum=0;
    printf("\n enter a value:");
    scanf("%d" , &n);
    while(i<=n){
        sum=sum+pow(i,2);
        i=i+2;
    }
    printf("\n The sum =%d" , sum);
}
```

Output:-



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 15_sum_of_series.c -o 15_sum_of_series }; if ($?) { .\15_sum_of_series }

enter a value:1234

The sum =313179945
PS D:\Codes\C Programming\Lab copy 2> |
```

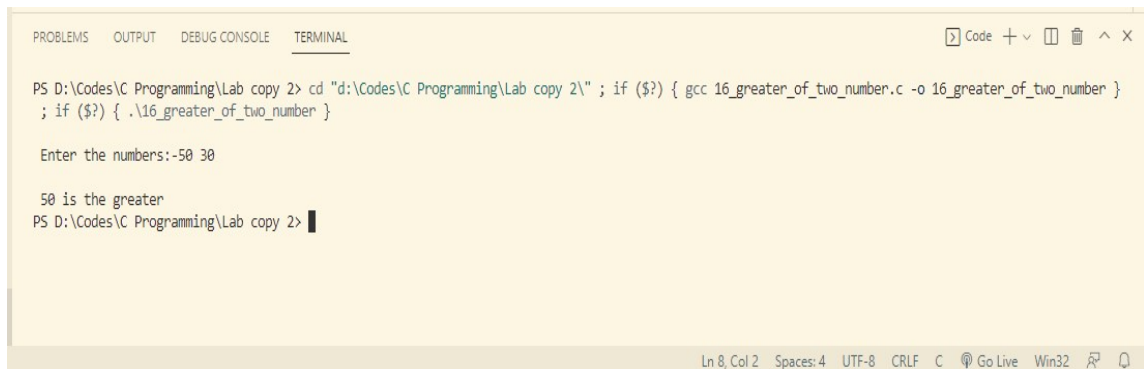
Ln 12, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32 🔍 🔔

20. Write a program in C to calculate the greater of two numbers using ternary operator:-

Code:-

```
#include<stdio.h>
#include<math.h>
int main(){
    int a,b;
    printf("\n Enter the numbers:-");
    scanf("%d %d" , &a ,&b);
    (a>b)? printf("\n %d is the greater" , a) : printf("\n %d is greater" , b);
}
```

Output:-

A screenshot of a code editor window with a terminal view. The terminal shows the execution of a C program. The prompt is 'PS D:\Codes\C Programming\Lab copy 2>'. The user enters 'cd "d:\Codes\C Programming\Lab copy 2\" ; if (\$?) { gcc 16_greater_of_two_number.c -o 16_greater_of_two_number } ; if (\$?) { .\16_greater_of_two_number }'. The program then prompts 'Enter the numbers:-' and the user enters '50 30'. The output is '50 is the greater'. The terminal status bar at the bottom shows 'Ln 8, Col 2', 'Spaces: 4', 'UTF-8', 'CRLF', 'C', 'Go Live', 'Win32', and icons for a refresh and a bell.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code + - - - - - ^ X

PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 16_greater_of_two_number.c -o 16_greater_of_two_number }
; if ($?) { .\16_greater_of_two_number }

Enter the numbers:-50 30

50 is the greater
PS D:\Codes\C Programming\Lab copy 2> █

Ln 8, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32
```

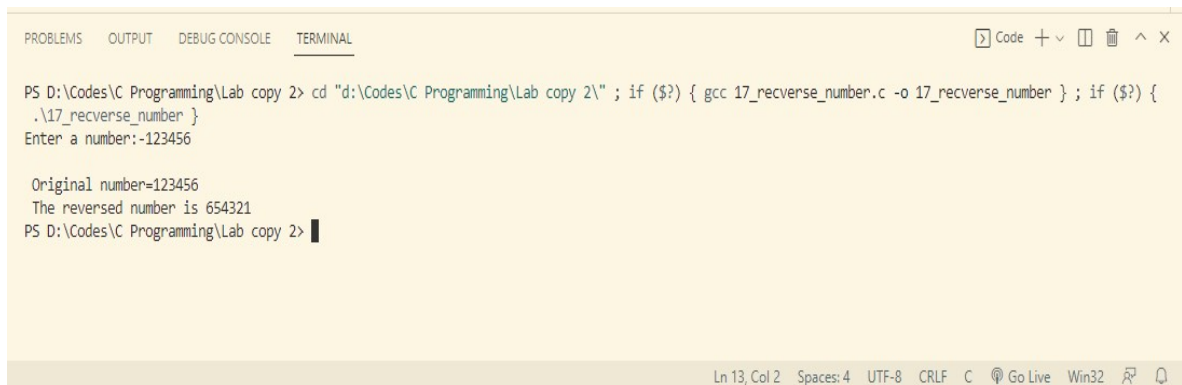
21. Write a C program to reverse a number:-

Code:-

```
#include<stdio.h>

int main(){
    int num ,rem, reverse=0;
    printf("Enter a number:-");
    scanf("%d" , &num);
    printf("\n Original number=%d" , num);
    while(num!=0){
        rem=num%10;
        reverse=reverse*10+rem;
        num=num/10;
    }
    printf("\n The reversed number is %d" , reverse);
}
```

Output:-



The screenshot shows a terminal window with a yellow background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL', with 'TERMINAL' being the active tab. The terminal content shows the command prompt 'PS D:\Codes\C Programming\Lab copy 2>' followed by a command to compile and run a C program: 'cd "d:\Codes\C Programming\Lab copy 2\" ; if (\$?) { gcc 17_reverse_number.c -o 17_reverse_number } ; if (\$?) { .\17_reverse_number }'. The program's output is displayed: 'Enter a number:-123456', 'Original number=123456', and 'The reversed number is 654321'. The prompt returns to 'PS D:\Codes\C Programming\Lab copy 2>'. At the bottom right, the status bar shows 'Ln 13, Col 2', 'Spaces: 4', 'UTF-8', 'CRLF', 'C', 'Go Live', 'Win32', and some icons.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code + - - - - X

PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 17_reverse_number.c -o 17_reverse_number } ; if ($?) {
.\17_reverse_number }
Enter a number:-123456

Original number=123456
The reversed number is 654321
PS D:\Codes\C Programming\Lab copy 2>

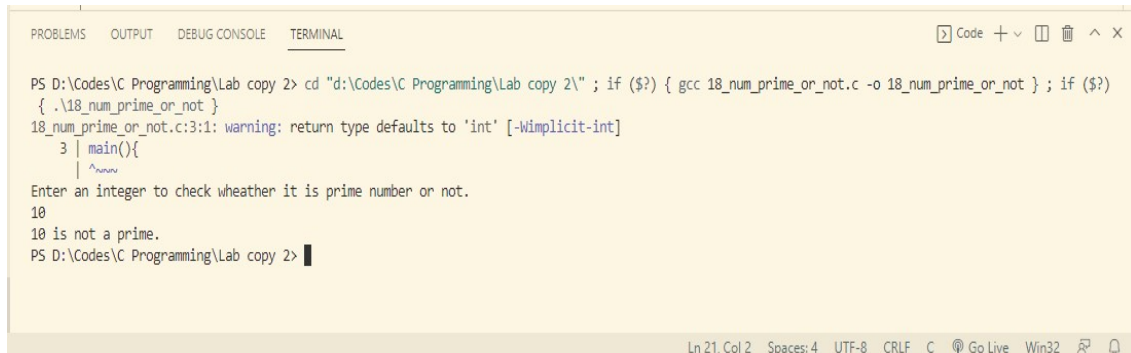
Ln 13, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32
```

22. Write a C program to find whether the number is prime or not using function.

Code:-

```
#include<stdio.h>
int check_prime(int);
main(){
    int n,result;
    printf("Enter an integer to check whether it is prime number or not.\n");
    scanf("%d", &n);
    result=check_prime(n);
    if(result==1)
        printf("%d is a prime.\n",n);
    else
        printf("%d is not a prime.\n",n);
    return 0;
}
int check_prime(int a){
    int c;
    for(c=2; c<=a-1; c++){
        if(a%c==0)
            return 0;
    }
    return 1;
}
```

Output:-



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Codes\C Programming\Lab copy 2> cd "d:\Codes\C Programming\Lab copy 2\" ; if ($?) { gcc 18_num_prime_or_not.c -o 18_num_prime_or_not } ; if ($?)
{ .\18_num_prime_or_not }
18_num_prime_or_not.c:3:1: warning: return type defaults to 'int' [-Wimplicit-int]
   3 | main(){
     | ^~~~~
Enter an integer to check whether it is prime number or not.
10
10 is not a prime.
PS D:\Codes\C Programming\Lab copy 2> █
```

Ln 21, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32