

# **CO24497: PROGRAMMING PRACTICES**

## **Lab Journals**



**Session 2023-2024**

**Submitted To:**

**Surendra Gupta**

**Ms. Mamta Gupta**

**Ms. Saloni Jain**

**Ms. Nikita Tiwari**

**Submitted By:**

**Mohan Manjhi**

**0801CS233D05**

**Computer Engineering Department**

**SHRI G. S. INSTITUTE OF TECHNOLOGY &  
SCIENCE, INDORE**

S. No.	Assignment Name	Submission Date	Page No.	Remark
1	<p><b>Programming Assignments</b></p> <ol style="list-style-type: none"> <li>1. Write a C program to print the following characters in reverse.</li> <li>2. Write a C program to compute the perimeter and area of a rectangle with a height and width.</li> <li>3. Write a C program to read an amount (integer value) and break the amount into the smallest possible number of bank notes.</li> <li>4. Write a C program that read 5 numbers and sum of all odd values between them.</li> <li>5. Write a C program that finds all the divisors of an integer.</li> <li>6. Write a C program that swaps two numbers without using a third variable.</li> <li>7. Write a C program to reverse and print a given number.</li> <li>8. Write a C program to display the sum of series <math>1 + 1/2 + 1/3 + \dots + 1/n</math>.</li> <li>9. WAP to find number is even or odd?</li> <li>10. WAP to convert decimal to binary.</li> <li>11. WAP to reverse a string.</li> <li>12. <math>2^{15} = 32768</math> and the sum of its digits is <math>3 + 2 + 7 + 6 + 8 = 26</math>.</li> <li>13. Write a Program to print Binary Equivalent of an Integer using Recursion.</li> <li>14. Write a program to calculate power of any number.</li> <li>15. Write a program to print if a number is a palindrome number or not.</li> <li>16. Write a program to print Armstrong series.</li> <li>17. Write a program to remove the all-</li> </ol>		4-30	

	<p>duplicates numbers from the given array.</p> <p>18. WAP to print diamond pattern in c.</p> <p>19. Write a program in C to sort elements of an array in ascending order.</p> <p>20. Write a program to check if a year is leap year or not.</p> <p>21. Write a program to print Fibonacci series.</p> <p>22. Write a C program to find the largest element in an array.</p> <p>23. WAP to find the length of a string without using strlen() function.</p> <p>24. Write a program to print Factorial of a number using Recursion?</p> <p>25. Write a c program to print multiplication of 2 matrixs.</p>			
<b>2</b>	<p><b>Skill based activities</b></p> <ul style="list-style-type: none"> <li>1. Coding Standards</li> <li>2. Version controlling</li> <li>3. Python</li> <li>4. Profiling</li> <li>5. Debugging</li> <li>6. LaTex</li> <li>7. Infrastrutures</li> <li>8. Linux</li> </ul>		<b>31-95</b>	
<b>3</b>	<p><b>Mini Project</b></p> <ul style="list-style-type: none"> <li>1. Overview</li> <li>2. Code</li> <li>3. Latex</li> </ul>		<b>96-103</b>	

# **25 Programming Practice Questions In C**

## 25 Programming Practice Questions

1. Write a C program to print the following characters in reverse.

Test Characters: 'X', 'M', 'L'

Expected Output:

The reverse of XML is LMX

```
#include <stdio.h>

int main()
{
    // Declare and initialize character variables
    char char1 = 'X';
    char char2 = 'M';
    char char3 = 'L';

    // Print the original and reversed characters
    printf("The reverse of %c%c%c is %c%c%c\n",
        char1, char2, char3,
        char3, char2, char1);

    return(0);
}
```

Output:

```
/tmp/N19mGhua6w.o
The reverse of XML is LMX
```

2. Write a C program to compute the perimeter and area of a rectangle with a height and width.

```
#include <stdio.h>
int main() {
    int width,height,area,perimeter;
    printf("Enter the height of rectangle");
    scanf("%d",&height);
    printf("Enter the width of rectangle");
    scanf("%d",&widht);
    perimeter = 2*(height + width);
    printf("Perimeter of the rectangle = %d inches\n", perimeter);

    area = height * width;
    printf("Area of the rectangle = %d square inches\n", area);

    return(0);
}
```

#### Output

```
/tmp/N19mGhua6w.o
Enter the height of rectangle : 5
Enter the width of rectangle : 7
Perimeter of the rectangle = 24 inches
Area of the rectangle = 35 square inches
```

Q3. Write a C program to read an amount (integer value) and break the amount into the smallest possible number of bank notes.

Test Data :

Input the amount: 375

Expected Output:

There are:

3 Note(s) of 100.00  
1 Note(s) of 50.00  
1 Note(s) of 20.00  
0 Note(s) of 10.00  
1 Note(s) of 5.00  
0 Note(s) of 2.00  
0 Note(s) of 1.00

```
#include <stdio.h>
int main() {
    int amt, total;

    printf("Input the amount: ");
    scanf("%d",&amt);

    total = (int)amt/100;
    printf("There are:\n");
    printf("%d Note(s) of 100.00\n", total);
    amt = amt-(total*100);

    total = (int)amt/50;
    printf("%d Note(s) of 50.00\n", total);
    amt = amt-(total*50);

    total = (int)amt/20;
    printf("%d Note(s) of 20.00\n", total);
    amt = amt-(total*20);

    total = (int)amt/10;
    printf("%d Note(s) of 10.00\n", total);
    amt = amt-(total*10);

    total = (int)amt/5;
    printf("%d Note(s) of 5.00\n", total);
    amt = amt-(total*5);
```

```
total = (int)amt/2;  
printf("%d Note(s) of 2.00\n", total);  
amt = amt-(total*2);  
  
total = (int)amt/1;  
printf("%d Note(s) of 1.00\n", total);  
  
return 0;  
}
```

### Output

```
/tmp/N19mGhua6w.o  
Input the amount: 375  
There are:  
3 Note(s) of 100.00  
1 Note(s) of 50.00  
1 Note(s) of 20.00  
0 Note(s) of 10.00  
1 Note(s) of 5.00  
0 Note(s) of 2.00  
0 Note(s) of 1.00
```

4. Write a C program that read 5 numbers and sum of all odd values between them.

```
#include <stdio.h>
int main() {
    int j, numbers[5], total=0;
    printf("\nInput the first number: ");
    scanf("%d", &numbers[0]);
    printf("\nInput the second number: ");
    scanf("%d", &numbers[1]);
    printf("\nInput the third number: ");
    scanf("%d", &numbers[2]);
    printf("\nInput the fourth number: ");
    scanf("%d", &numbers[3]);
    printf("\nInput the fifth number: ");
    scanf("%d", &numbers[4]);
    for(j = 0; j < 5; j++) {
        if((numbers[j]%2) != 0)
        {
            total += numbers[j];
        }
    }
    printf("\nSum of all odd values: %d", total);
    return 0;
}
```

Output

```
/tmp/N19mGhua6w.o
Input the first number: 10
Input the second number: 25
Input the third number: 12
Input the fourth number: 16
Input the fifth number: 20
Sum of all odd values: 25
```

5. Write a C program that finds all the divisors of an integer.

Test Data: Input an integer: 45

Expected Output: All the divisor of 45 are : 1 3 5 9 15 45

```
#include <stdio.h>
int main() {
    int x, i;

    // Get an integer input from the user
    printf("\nInput an integer: ");
    scanf("%d", &x);

    // Print all the divisors of x
    printf("All the divisors of %d are: ", x);
    for(i = 1; i <= x; i++) {
        if((x%i) == 0){
            printf("\n%d", i);
            printf("\n");
        }
    }

    return 0;
}
```

#### Output

```
/tmp/N19mGhua6w.o
Input an integer: 78
All the divisors of 78 are:
1
2
3
6
13
26
39
78
```

6. Write a C program that swaps two numbers without using a third variable.

Input value for x & y:

Before swapping the value of x & y: 5 7

After swapping the value of x & y: 7 5

```
#include<stdio.h>
int main() {
    int x, y;

    // Prompt user to input values for x and y
    printf("Input value for x & y: \n");
    scanf("%d%d",&x,&y);

    // Display the values of x and y before swapping
    printf("Before swapping the value of x & y: %d %d",x,y);

    // Swap the values of x and y using arithmetic operations
    x = x + y;
    y = x - y;
    x = x - y;

    // Display the values of x and y after swapping
    printf("\nAfter swapping the value of x & y: %d %d",x,y);

    return 0;
}
```

#### Output

```
/tmp/N19mGhua6w.o
Input value for x & y:
5
10
Before swapping the value of x & y: 5 10
After swapping the value of x & y: 10 5|
```

7. Write a C program to reverse and print a given number.

Input a number:

The original number = 234

The reverse of the said number = 432

```
#include<stdio.h>
int main() {
    int num, x, r_num = 0;

    printf("Input a number: ");
    scanf("%d", &num);

    printf("\nThe original number = %d", num);

    // Reverse the digits of the number
    while (num >= 1) {
        x = num % 10;
        r_num = r_num * 10 + x;
        num = num / 10;
    }

    // Print the reverse of the number
    printf("\nThe reverse of the said number = %d", r_num);

    return 0;
}
```

#### Output

```
/tmp/N19mGhua6w.o
Input a number: 125
The original number = 125
The reverse of the said number = 521
```

8. Write a C program to display the sum of series  $1 + 1/2 + 1/3 + \dots + 1/n$ .

```
#include<stdio.h>
int main() {
    int num, i, sum = 0;
    printf("Input any number: ");
    scanf("%d", &num);
    printf("1 + ");

    for(i = 2; i <= num - 1; i++)
        printf(" 1/%d +", i);

    for(i = 1; i <= num; i++)
        sum = sum + i;

    printf(" 1/%d", num);

    printf("\nSum = 1/%d", sum + 1/num);

    return 0;
}
```

### Output

```
/tmp/N19mGhua6w.o
Input any number: 8
1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + 1/7 + 1/8
Sum = 1/36
```

9. WAP to find number is even or odd?

```
#include <stdio.h>
int main() {
    int value;
    scanf("%d",&value);
    if(value%2==0){
        printf("Even");
    }else{
        printf("Odd");
    }
    return 0;
}
```

Output

```
/tmp/N19mGhua6w.o
11
Odd
```

10. WAP to convert decimal to binary.

```
#include<stdio.h>
int main(){
int a[10],n,i;
printf("Enter the number to convert: ");
scanf("%d",&n);
for(i=0;n>0;i++)
{
a[i]=n%2;
n=n/2;
}
printf("\nBinary of Given Number is=");
for(i=i-1;i>=0;i--)
{
printf("%d",a[i]);
}
return 0;
}
```

Output

```
/tmp/N19mGhua6w.o
Enter the number to convert: 8
Binary of Given Number is=1000
```

11. WAP to reverse a string.

```
# include <stdio.h>
void reverse(char *str)
{
    if (*str)
    {
        reverse(str + 1);
        printf("%c", *str);
    }
}
int main()
{
    char a[] = "NAHOM";
    reverse(a);
    return 0;
}
```

Output

```
/tmp/viVof63ArX.o
MOHAN|
```

12.  $2^{15} = 32768$  and the sum of its digits is  $3 + 2 + 7 + 6 + 8 = 26$ .

```
#include <stdio.h>
#include <math.h>
int main()
{
    int n;
    printf("Enter the value of 2^x ");
    scanf("%d",&n);
    int b=pow(2,n);
    printf(" %d\n",b);
    printf("Sum of all digit is %d ", getSum(b));
    return 0;
}
int getSum(int b)
{
    int sum = 0;
    while (b != 0) {
        sum = sum + b % 10;
        b = b / 10;
    }
    return sum;
}
```

#### Output

```
/tmp/KbDszU8wY2.o
Enter the value of 2^x 5
32
Sum of all digit is 5 |
```

13. Write a Program to print Binary Equivalent of an Integer using Recursion.

```
#include <stdio.h>
int binary_conversion(int);
int main()
{
    int num, bin;
    printf("Enter a decimal number: ");
    scanf("%d", &num);
    bin = binary_conversion(num);
    printf("The binary equivalent of %d is %d\n", num, bin);
}
int binary_conversion(int num)
{
    if (num == 0)
    {
        return 0;
    }
    else
    {
        return (num % 2) + 10 * binary_conversion(num / 2);
    }
}
```

Output

```
/tmp/KbDszU8wY2.o
Enter a decimal number: 25
The binary equivalent of 25 is 11001
```

14. Write a program to calculate power of any number.

```
#include <stdio.h>
long power(int x, unsigned n)
{
    long long pow = 1;
    for (int i = 0; i < n; i++) {
        pow = pow * x;
    }

    return pow;
}
int main(void)
{
    int x;
    printf("Enter value of x:\n");
    scanf("%d",&x);
    unsigned n;
    printf("Enter value of n:\n");
    scanf("%d",&n);
    int result = power(x, n);
    printf("%d", result);
    return 0;
}
```

#### Output

```
/tmp/KbDszU8wY2.o
Enter value of x:
5
Enter value of n:
3
125
```

15. Write a program to print if a number is a palindrome number or not.

```
#include <stdio.h>
int main()
{
    int original_number;
    printf("Enter value:\n");
    scanf("%d",&original_number);
    int reversed = 0;
    int num = original_number;
    while (num != 0) {
        int r = num % 10;
        reversed = reversed * 10 + r;
        num /= 10;
    }
    if (original_number == reversed) {
        printf(" Given number %d is a palindrome number",
               original_number);
    }
    else {
        printf(
            " Given number %d is not a palindrome number",
            original_number);
    }
    return 0;
}
```

### Output

```
/tmp/KbDszU8wY2.o
Enter value:
9
Given number 9 is a palindrome number
```

16. Write a program to print Armstrong series.

```
#include <stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int temp = n;
    int p = 0;

    while (n > 0) {
        int rem = n % 10;
        p = (p) + (rem * rem * rem);
        n = n / 10;
    }

    if (temp == p) {
        printf("Yes. It is Armstrong No.");
    }
    else {
        printf("No. It is not an Armstrong No.");
    }
    return 0;
}
```

Output

```
/tmp/KbDszU8wY2.o
25
No. It is not an Armstrong No.
```

17. Write a program to remove the all-duplicates numbers from the given array.

```
#include <stdio.h>
int main()
{
    int n, count = 0;
    printf("Enter number of elements in the array: ");
    scanf("%d", &n);
    int arr[n], temp[n];
    if(n==0){
        printf("No element inside the array.");
        exit(0);
    }
    printf("Enter elements in the array: ");
    for (int i = 0; i < n; i++){
        scanf("%d", &arr[i]);
    }
    printf("\nArray Before Removing Duplicates: ");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    for (int i = 0; i < n; i++){
        int j;
        for (j = 0; j < count; j++){
            if (arr[i] == temp[j])
                break;
        }
        if (j == count){
            temp[count] = arr[i];
            count++;
        }
    }
    printf("\nArray After Removing Duplicates: ");
    for (int i = 0; i < count; i++)
        printf("%d ", temp[i]);
    return 0;
}
```

#### Output

```
/tmp/KbDszU8wY2.o
Enter number of elements in the array: 5
Enter elements in the array: 2 3 9 2 9 2
Array Before Removing Duplicates: 2 3 9 2 9
Array After Removing Duplicates: 2 3 9 |
```

18. WAP to print diamond pattern in c.

```
#include <stdio.h>
int main()
{
    int n, c, k;
    printf("Enter number of rows\n");
    scanf("%d", &n);
    for (k = 1; k <= n; k++)
    {
        for (c = 1; c <= n-k; c++)
            printf(" ");
        for (c = 1; c <= 2*k-1; c++)
            printf("*");
        printf("\n");
    }
    for (k = 1; k <= n - 1; k++)
    {
        for (c = 1; c <= k; c++)
            printf(" ");
        for (c = 1 ; c <= 2*(n-k)-1; c++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

### Output

```
/tmp/KbDszU8wY2.o
Enter number of rows : 4
*
 ***
 *****
 *****
 ***
 *
 *
```

19. Write a program in C to sort elements of an array in ascending order.

Test Data :

Input the size of array : 5

Input 5 elements in the array :

element - 0 : 2

element - 1 : 7

element - 2 : 4

element - 3 : 5

element - 4 : 9

Expected Output :

Elements of array in sorted ascending order:

2 4 5 7 9

```
#include <stdio.h>
void main(){
    int arr1[100];
    int n, i, j, tmp;
    printf("\n\nsort elements of array in ascending order :\n ");
    printf("Input the size of array : ");
    scanf("%d", &n);
    printf("Input %d elements in the array :\n",n);
    for(i=0;i<n;i++){
        printf("element - %d : ",i);
        scanf("%d",&arr1[i]);
    }
    for(i=0; i<n; i++){
        for(j=i+1; j<n; j++){
            if(arr1[j] <arr1[i]){
                tmp = arr1[i];
                arr1[i] = arr1[j];
                arr1[j] = tmp;
            }
        }
    }
    printf("\nElements of array in sorted ascending order:\n");
    for(i=0; i<n; i++){
        printf("%d ", arr1[i]);
    }
    printf("\n\n");}
```

### Output

```
/tmp/KbDszU8wY2.o
sort elements of array in ascending order :
Input the size of array : 5
Input 5 elements in the array :
element - 0 : 2
element - 1 : 7
element - 2 : 4
element - 3 : 5
element - 4 : 9
Elements of array in sorted ascending order:
2 4 5 7 9
```

20. Write a program to check if a year is leap year or not.

```
#include <stdio.h>
int main() {
    int year;
    printf("Enter a year: ");
    scanf("%d", &year);
    if (year % 400 == 0) {
        printf("%d is a leap year.", year);
    }
    else if (year % 100 == 0) {
        printf("%d is not a leap year.", year);
    }
    else if (year % 4 == 0) {
        printf("%d is a leap year.", year);
    }
    else {
        printf("%d is not a leap year.", year);
    }
    return 0;
}
```

### Output

```
/tmp/KbDszU8wY2.o
Enter a year: 2020
2020 is a leap year.|
```

21. Write a program to print Fibonacci series.

```
#include <stdio.h>
int prev1 = 1;
int prev2 = 0;
void fib(int n)
{
    if (n < 3) {
        return;
    }
    int fn = prev1 + prev2;
    prev2 = prev1;
    prev1 = fn;
    printf("%d ", fn);
    return fib(n - 1);
}
void printFib(int n)
{
    if (n < 1) {
        printf("Invalid number of terms\n");
    }
    else if (n == 1) {
        printf("%d ", 0);
    }
    else if (n == 2) {
        printf("%d %d", 0, 1);
    }
    else {
        printf("%d %d ", 0, 1);
        fib(n);
    }
    return;
}
int main()
{
    int n;
    printf("Enter value:\n");
    scanf("%d",&n);
    printFib(n);
    return 0;
}
```

### Output

```
/tmp/KbDszU8wY2.o
Enter value:
11
0 1 1 2 3 5 8 13 21 34 55
```

22. Write a C program to find the largest element in an array.

```
#include <stdio.h>
int largest(int arr[], int n)
{
    int i;
    int max = arr[0];
    for (i = 1; i < n; i++)
        if (arr[i] > max)
            max = arr[i];
    return max;
}
int main()
{
    int arr[] = { 10, 324, 45, 90, 9808 };
    int n = sizeof(arr) / sizeof(arr[0]);
    printf("Largest in given array is %d", largest(arr, n));
    return 0;
}
```

### Output

```
/tmp/KbDszU8wY2.o
Largest in given array is 9808
```

23. WAP to find the length of a string without using strlen() function.

```
#include <stdio.h>
int main() {
    char s[] = "Hello my name is Mohan";
    int i;
    for (i = 0; s[i] != '\0'; ++i);
    printf("Length of the string: %d", i);
    return 0;
}
```

Output

```
/tmp/KbDszU8wY2.o
Length of the string: 22
```

24. Write a program to print Factorial of a number using Recursion?

```
#include<stdio.h>
long factorial(int n) {
    if (n == 0)
        return 1;
    else
        return(n * factorial(n-1));
}
void main() {
    int number;
    long fact;
    printf("Enter a number: ");
    scanf("%d", &number);
    fact = factorial(number);
    printf("Factorial of %d is %ld\n", number, fact);
    return 0;
}
```

Output

```
/tmp/KbDszU8wY2.o
Enter a number: 5
Factorial of 5 is 120
```

25. Write a c program to print multiplication of 2 matrices.

```
#include<stdio.h>
int main(){
int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;

printf("enter the number of row=");
scanf("%d",&r);

printf("enter the number of column=");
scanf("%d",&c);

printf("enter the first matrix element=\n");

for(i=0;i<r;i++){
    for(j=0;j<c;j++){
        scanf("%d",&a[i][j]);
    }
}

printf("enter the second matrix element=\n");

for(i=0;i<r;i++){
    for(j=0;j<c;j++)
    {
        scanf("%d",&b[i][j]);
    }
}

printf("multiply of the matrix=\n");

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

```
    }

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

return 0;
}
```

### Output

```
/tmp/JxvTxitre5.o
enter the number of row : 2
enter the number of column : 2
enter the first matrix element :
2
5
7
3
5
enter the second matrix element :
2
5
9
6
multiply of the matrix :
67  52
51  45
```

# **MY CODING STANDARDS**

## **MY CODING STANDARDS**

Listed below are the coding practices that I will follow:

### **Variable Naming**

snake case letter will be used to denote the variable name, and first letter always will be start with capital letter and in my variable naming I use short form of data type to easily identify variable data type at any place I use “In” for int, “Ch” for char “Fl” for float and so on. Data type short name use in starting of variable with underscore to define a variable name.

Eg: int In\_age; char Ch\_letter; float Fl\_value; double D\_value;

### **Function Naming:**

camel case will be used to define functions, i.e the 1<sup>st</sup>letter will be small and the 1<sup>st</sup>letter of the next words will be capital.

Eg: void addFun(); int addFun();

### **Indentations:**

space of one tab will be left when there is the use of ‘{’, plus any other time there is a need to separate the coming part from the general flow of control like calling a function. And use the inline comment in end of the control statement that can easy to understand where any control statement is ending.

Eg: for(int i=0; i<10; i++){  
int In\_a=0; // initialize a  
a=i; //assign the value  
} //end of for loop

### **4. Comments:**

In my coding standard I use single line comment use to define the end of loop and end of the statement braces that help to understand that where is a statement ending and I will use multi-line comments to define a control statement , function and methods because it is very helpful to understand the work of any control statement, function and method.

**Eg:**

```
// create a function to calculate sum of a  
and b int myFunction(int a, int b) { int In_a1  
= a; int In_b2 = b; return a1+b2;  
}  
int main() { int sum = myFunction(10,20); //  
call the function printf("%d",sum); return 0;  
}
```

**5. Sequencing:**

the sequencing of the program will be as follows:

1. The copyright statement
2. #include commands
3. Function if need.
4. main function-
5. variable declaration and use the statements like control statement, conditional statement.

This sequence will ensure a uniform approach to the program that can be easily understood in later stages.

**\*Assignment\***

# **Version Controlling**

**(Git Assignment)**

## Step 1. Create a folder Programming lab.



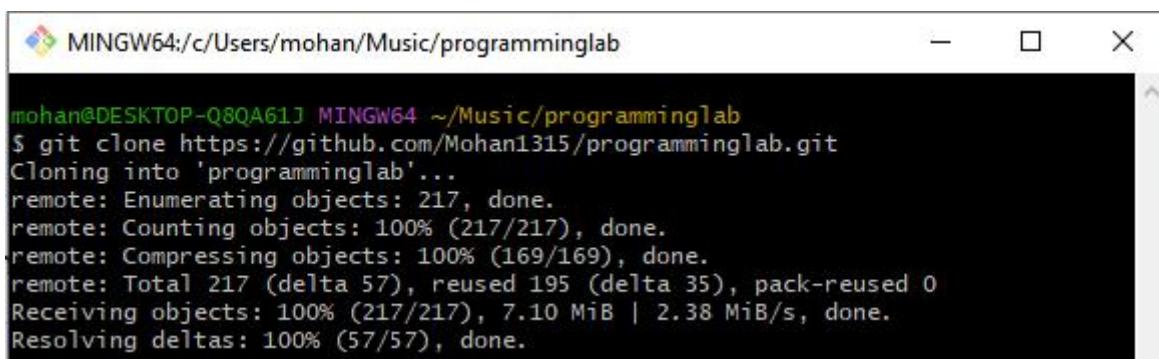
```
MINGW64:/c/Users/mohan/Music/programminglab
mohan@DESKTOP-Q8QA61J MINGW64 ~/Music
$ mkdir programminglab

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music
$ ls
Playlists/ desktop.ini programminglab/

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music
$ cd programminglab

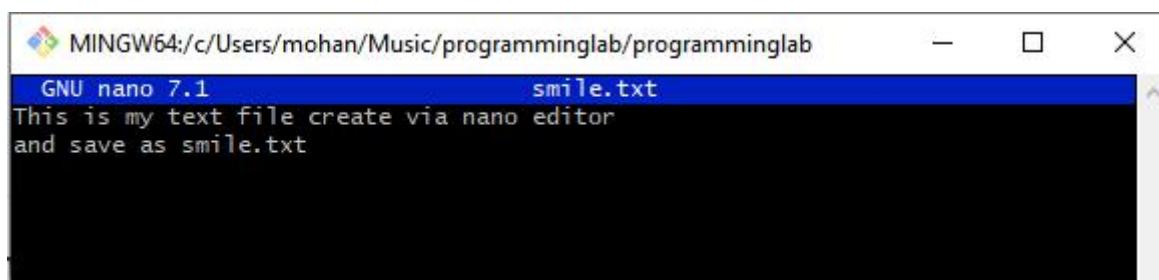
mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab
$ |
```

## Step 2. Cloning the existed Repository from My github



```
MINGW64:/c/Users/mohan/Music/programminglab
mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab
$ git clone https://github.com/Mohan1315/programminglab.git
Cloning into 'programminglab'...
remote: Enumerating objects: 217, done.
remote: Counting objects: 100% (217/217), done.
remote: Compressing objects: 100% (169/169), done.
remote: Total 217 (delta 57), reused 195 (delta 35), pack-reused 0
Receiving objects: 100% (217/217), 7.10 MiB | 2.38 MiB/s, done.
Resolving deltas: 100% (57/57), done.
```

## Step 3. Creating a new Text file



```
MINGW64:/c/Users/mohan/Music/programminglab/programminglab
GNU nano 7.1           smile.txt
This is my text file create via nano editor
and save as smile.txt
```

## Step 4. Adding it to Staging area and commit then push.

```
MINGW64:/c/Users/mohan/Music/programminglab/programminglab
mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    smile.txt

nothing added to commit but untracked files present (use "git add" to track)

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (master)
$ git add .
warning: in the working copy of 'smile.txt', LF will be replaced by CRLF the next time Git touches it

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (master)
$ git commit -m "smile.txt file create"
[master e0e4a44] smile.txt file create
 1 file changed, 2 insertions(+)
 create mode 100644 smile.txt

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 339 bytes | 339.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Mohan1315/programminglab.git
  f609039..e0e4a44  master -> master
```

## Before Pushing

 Mohan1315 new update	f609039 53 minutes ago	 29 commits
 .vscode	delete some file	last week
 Shoping Cart project	delete some file	last week
 movie ticket project	new update	53 minutes ago
 Demo.zip	delete some file	last week
 ProgrammingLab.zip	delete some file	last week
 mohan.txt	initial commit	last week
 program1.py	delete some file	last week

## After Pushing

	Mohan1315 smile.txt file create	e0e4a44 2 minutes ago	⌚ 30 commits
	.vscode delete some file		last week
	Shoping Cart project delete some file		last week
	movie ticket project new update		1 hour ago
	Demo.zip delete some file		last week
	ProgrammingLab.zip delete some file		last week
	mohan.txt initial commit		last week
	program1.py delete some file		last week
	smile.txt smile.txt file create	2 minutes ago	

## Step 5. Switcing to New Branch.

```

MINGW64:/c/Users/mohan/Music/programminglab/programminglab

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (NewMohan)
$ git branch
  Mohan
* NewMohan
  master

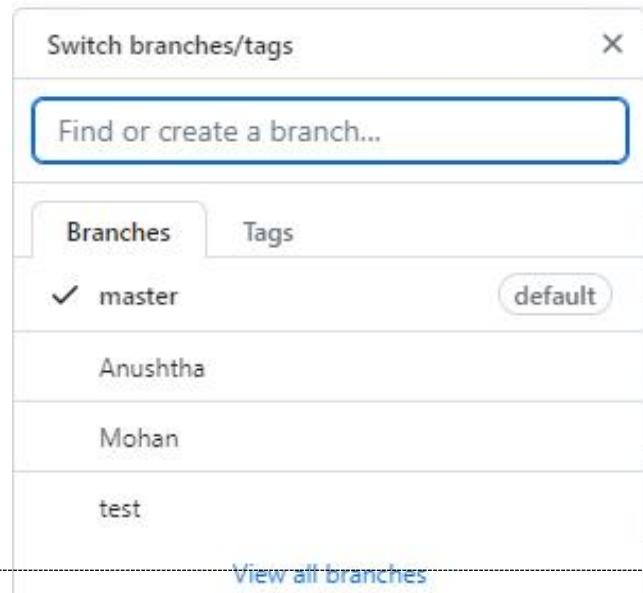
mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (NewMohan)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (master)
$ git merge NewMohan
Updating e0e4a44..ebf6a75
Fast-forward
  cry.txt | 2 ++
  1 file changed, 2 insertions(+)
  create mode 100644 cry.txt

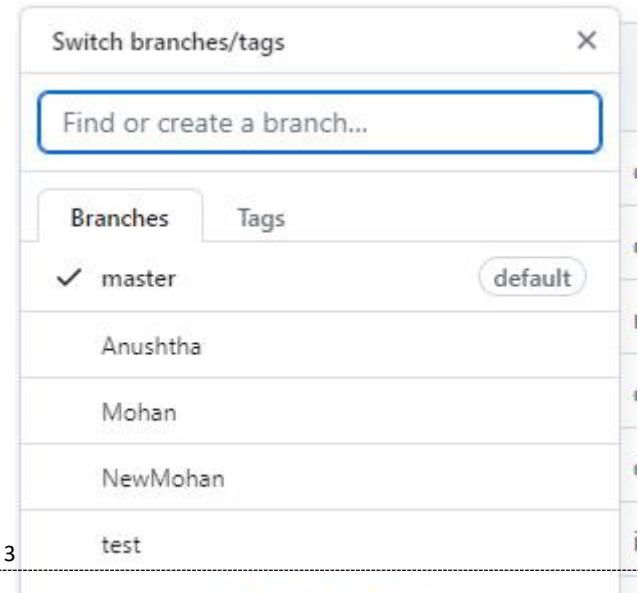
mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Mohan1315/programminglab.git
  e0e4a44..ebf6a75  master -> master

```

Before adding New branch



After pushing new Branch



## Step 6. Creating File on new Branch.

```
MINGW64:/c/Users/mohan/Music/programminglab/program... - X
GNU nano 7.1      cry.txt
this is my new text file for new branch.
```

## Pushing File in Git through new Branch

```
MINGW64:/c/Users/mohan/Music/programminglab/programminglab - X
$ git add cry.txt
warning: in the working copy of 'cry.txt', LF will be replaced by CRLF
the next time Git touches it

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (NewMohan)
$ git status
On branch NewMohan
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   cry.txt

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (NewMohan)
$ git commit -m "adding file in new branch"
[NewMohan ebf6a75] adding file in new branch
 1 file changed, 2 insertions(+)
 create mode 100644 cry.txt

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (NewMohan)
$ git push origin NewMohan
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 309 bytes | 309.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Mohan1315/programminglab.git
  e0e4a44..ebf6a75  NewMohan -> NewMohan
```

```
MINGW64:/c/Users/mohan/Music/programminglab/programminglab
mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (NewMohan)
$ git branch
  Mohan
* NewMohan
  master

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (NewMohan)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (master)
$ git merge NewMohan
Updating e0e4a44..ebf6a75
Fast-forward
  cry.txt | 2 ++
  1 file changed, 2 insertions(+)
  create mode 100644 cry.txt

mohan@DESKTOP-Q8QA61J MINGW64 ~/Music/programminglab/programminglab (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Mohan1315/programminglab.git
  e0e4a44..ebf6a75  master -> master
```

## Before Merging

 Mohan1315	smile.txt file create	e0e4a44 34 minutes ago	 30 commits
 .vscode	delete some file		last week
 Shoping Cart project	delete some file		last week
 movie ticket project	new update		1 hour ago
 Demo.zip	delete some file		last week
 ProgrammingLab.zip	delete some file		last week
 mohan.txt	initial commit		last week
 program1.py	delete some file		last week
 smile.txt	smile.txt file create		34 minutes ago

## After Merging

	Mohan1315 adding file in new branch	ebf6a75 5 minutes ago	 31 commits
	.vscode delete some file		last week
	Shoping Cart project delete some file		last week
	movie ticket project new update		1 hour ago
	Demo.zip delete some file		last week
	ProgrammingLab.zip delete some file		last week
	cry.txt adding file in new branch		5 minutes ago
	mohan.txt initial commit		last week
	program1.py delete some file		last week
	smile.txt smile.txt file create		37 minutes ago

# **Python Assignment**

1. Using Pandas read any CSV file and display first five rows of that file.

In [1]:

```
import pandas as pd
data = pd.read_csv('employees.csv')
print(data)
```

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
0	198	Donald	OConnell	DOCONNEL	650.507.9833	21-Jun-07
1	199	Douglas	Grant	DGRANT	650.507.9844	13-Jan-08
2	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-Sep-03
3	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-Feb-04
4	202	Pat	Fay	PFAY	603.123.6666	17-Aug-05

	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
0	SH_CLERK	2600	-	124	50
1	SH_CLERK	2600	-	124	50
2	AD_ASST	4400	-	101	10
3	MK_MAN	13000	-	100	20
4	MK_REP	6000	-	201	20

2. Write a python program to print the contents of a directory using OS module. Search online for the function which does that.

In [10]:

```
import os
dir_path = "\\\Users\\mohan\\Downloads\\Python Assignment"
dir_contents = os.listdir(dir_path)

for item in dir_contents:
    print(item)
```

.ipynb\_checkpoints  
Date\_day.ipynb  
dic\_to\_xml.py  
dir\_path.py  
employees.csv  
image.py  
Os\_module.ipynb  
Q10.ipynb  
Q10.py  
Q8.py  
Q9.py  
random\_num.py  
read.py  
read\_csv.ipynb  
read\_csvFile.ipynb  
read\_file.ipynb  
symmetric\_diff.py  
tempCodeRunnerFile.py  
Untitled.ipynb  
validate\_num.py

3. Write a program to read image and find size and shape of image using external module.

In [\*]:

```
import cv2
image_path = '\\\\Users\\\\mohan\\\\Downloads\\\\Python Assignment\\\\images.png'
image = cv2.imread(image_path)

if image is not None:
    height, width, _ = image.shape
    num_channels = image.shape[2]

    print(f"Image Size (Width x Height): {width} x {height}")
    print(f"Number of Channels: {num_channels}")

    cv2.imshow('Image', image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
else:
    print("Error: Unable to load the image.")

Image Size (Width x Height): 225 x 225
Number of Channels: 3
```

In [ ]:

4. Please write a program to make a list of random numbers between 0 to 10 and output numbers from that list which is divisible by 5 or 7, using random module.

In [1]:

```
import random
random_numbers = [random.randint(0, 10) for _ in range(20)]
divisible_by_5_or_7 = [num for num in random_numbers if num % 5 == 0 or num % 7 == 0]

print("Random Numbers:", random_numbers)
print("Numbers Divisible by 5 or 7:", divisible_by_5_or_7)
```

Random Numbers: [0, 8, 6, 8, 8, 5, 0, 3, 6, 9, 2, 7, 6, 4, 7, 7, 5, 6, 0, 7]
Numbers Divisible by 5 or 7: [0, 5, 0, 7, 7, 5, 0, 7]

In [ ]:

## 5. Write a python program to convert dictionary to xml.

The screenshot shows a Jupyter Notebook interface with the title "jupyter Dict\_to\_xml". The code in cell [1] defines a function `dict_to_xml` that uses the `xml.etree.ElementTree` module to create an XML tree from a dictionary. It handles both dictionary values (which are converted to XML elements) and non-dictionary values (which are converted to text nodes). A sample dictionary `data` is provided, containing a single `person` entry with attributes name, age, city, and state. The resulting XML output is printed at the end of the code cell.

```
In [1]: import xml.etree.ElementTree as ET
def dict_to_xml(dictionary, root_name):
    root = ET.Element(root_name)
    for key, value in dictionary.items():
        if isinstance(value, dict):
            child = dict_to_xml(value, key)
            root.append(child)
        else:
            child = ET.Element(key)
            child.text = str(value)
            root.append(child)
    return root

data = {
    "person": {
        "name": "Mohan",
        "age": 20,
        "city": "Guna",
        "State": "M.p"
    }
}
root_element = dict_to_xml(data, "data")
xml_string = ET.tostring(root_element, encoding="unicode")

print(xml_string)
```

<data><person><name>Mohan</name><age>20</age><city>Guna</city><State>M.p</State></person></data>

## 6. Write a program to validate phone number.

The screenshot shows a Jupyter Notebook interface with the title "jupyter Validate\_phn\_number". The code in cell [1] defines a function `is_valid_phone_number` using regular expressions to check if a given phone number is valid (9 digits starting with 6, 7, 8, or 9). It then prompts the user to enter a phone number and prints whether it is valid or not based on the function's output.

```
In [1]: import re
def is_valid_phone_number(phone_number):

    pattern = r"^[6789]\d{9}$"

    if re.match(pattern, phone_number):
        return True
    else:
        return False

phone_number = input("Enter an Indian phone number: ")
phone_number = ''.join(filter(str.isdigit, phone_number))

if is_valid_phone_number(phone_number):
    print(f"{phone_number} is a valid Indian phone number.")
else:
    print(f"{phone_number} is not a valid Indian phone number.")

Enter an Indian phone number: 7879664973
7879664973 is a valid Indian phone number.
```

7. Given 2 sets of integers, M and N, print their symmetric difference in ascending order. The term symmetric difference indicates those values that exist in either M or N but do not exist in both.

Input Format :

The first line of input contains an integer, M.

The second line contains M space-separated integers.

The third line contains an integer, N.

The fourth line contains N space-separated integers.

Output Format :

Output the symmetric difference integers in ascending order, one per line.

Sample Input :

4

2 4 5 9

4

2 4 11 12

Sample Output:

5 9

11

12

The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** Jupyter Symetric\_diff Last Checkpoint: 2 minutes ago (unsaved changes) Trusted Python 3 (ipykernel) C Logout
- Toolbar:** File Edit View Insert Cell Kernel Widgets Help
- Cell 3:** In [3]:

```
m = int(input())
set_m = set(map(int, input().split()))

n = int(input())
set_n = set(map(int, input().split()))

symmetric_diff = set_m.symmetric_difference(set_n)
result = sorted(symmetric_diff)

for item in result:
    print(item)
```
- Output:** 4  
2 4 5 9  
4  
2 4 11 12  
5  
9  
11  
12
- Cell 4:** In [ ]:

8. Consider a list (list = []). You can perform the following commands:

1. insert i e: Insert integer e at position i.
2. print: Print the list.
3. remove e: Delete the first occurrence of integer e.
4. append e: Insert integer e at the end of the list.
5. sort: Sort the list.
6. pop: Pop the last element from the list.
7. reverse: Reverse the list.

Initialize your list and read in the value of n followed by n lines of commands where each command will be of the 7 types listed above. Iterate through each command in order and perform the corresponding operation on your list. Sample Input

12 insert 0 5

insert 1 10

insert 0 6 print

remove 6

append 9

append 1 sort

print pop

reverse print

Sample Output

[6, 5, 10]

[1, 5, 9, 10]

[9, 5, 1]

The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** jupyter List\_command Last Checkpoint: 9 minutes ago (unsaved changes) Trusted Python 3 (ipykernel)
- Toolbar:** File Edit View Insert Cell Kernel Widgets Help
- Cell Header:** In [ ]:
- Code Cell Content:** In [2]:

```
my_list = []
n = int(input())

for _ in range(n):
    command = input().split()

    if command[0] == "insert":
        i, e = int(command[1]), int(command[2])
        my_list.insert(i, e)
    elif command[0] == "print":
        print(my_list)
    elif command[0] == "remove":
        e = int(command[1])
        my_list.remove(e)
    elif command[0] == "append":
        e = int(command[1])
        my_list.append(e)
    elif command[0] == "sort":
        my_list.sort()
    elif command[0] == "pop":
        my_list.pop()
    elif command[0] == "reverse":
        my_list.reverse()
```
- Output Cell Content:** In [2]:

```
12
insert 0 5
insert 1 10
insert 0 6
print
[6, 5, 10]
remove 6
append 9
append 1
sort
print
[1, 5, 9, 10]
reverse
print
[9, 5, 1]
pop 2
```

9. You are given a date. Your task is to find what the day is on that date

**Input Format**

A single line of input containing the space separated month, day and year, respectively, in MM DD YYYY format.

**Constraints**

2000 < year < 3000

**Output Format**

Output the correct day in capital letters.

**Sample Input**

08 05 2015

**Sample Output**

WEDNESDAY

The screenshot shows a Jupyter Notebook interface. The top bar includes the logo, the title "jupyter Date\_day Last Checkpoint: 6 minutes ago (autosaved)", and a "Logout" button. The menu bar has options: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. Below the menu is a toolbar with icons for file operations like Open, Save, Run, and Cell Type. A status bar indicates "Trusted" and "Python 3 (ipykernel)". The main area contains a code cell labeled "In [1]:" with the following Python code:

```
from datetime import datetime
date_str = input("Enter the date (MM DD YYYY): ")
date = datetime.strptime(date_str, "%m %d %Y")
day_of_week = date.strftime("%A").upper()
print(day_of_week)
```

When run, the cell outputs:

```
Enter the date (MM DD YYYY): 10 26 2023
THURSDAY
```

Below the code cell is an empty input field labeled "In [ ]:".

10. Write a program which accepts a sequence of words separated by whitespace as input to print the words composed of digits only.

Example: If the following words are given as input to the program: 2 cats and 3 dogs.

Then, the output of the program should be: ['2', '3']

The screenshot shows a Jupyter Notebook interface. The top bar includes the logo, the title "jupyter Q10 Last Checkpoint: a minute ago (unsaved changes)", and a "Logout" button. The menu bar has options: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. Below the menu is a toolbar with icons for file operations like Open, Save, Run, and Cell Type. A status bar indicates "Trusted" and "Python 3 (ipykernel)". The main area contains a code cell labeled "In [1]:" with the following Python code:

```
import re
input_str = input("Enter a sequence of words: ")
digit_words = re.findall(r'\b\d+\b', input_str)
print(digit_words)
```

When run, the cell outputs:

```
Enter a sequence of words: My name is Mohan my fav number is 7 and 3 also
['7', '3']
```

Below the code cell is an empty input field labeled "In [ ]:".

# Profiling Assignment

**Q.1 Use gprof to analyze and if required optimize the below program:**

```
#include <stdio.h>
#include <stdlib.h>
// A function to generate random numbers and store them in an array
void generate_random_numbers(int* array, int size) {
    for (int i = 0; i < size; i++) {
        array[i] = rand() % 1000;
    }
}
// A function to find the sum of elements in an array
int find_sum(int* array, int size) {
    int sum = 0;
    for (int i = 0; i < size; i++) {
        sum += array[i];
    }
    return sum;
}

// A function to find the product of elements in an array
int find_product(int* array, int size) {
    int product = 1;
    for (int i = 0; i < size; i++) {
        product *= array[i];
    }
    return product;
}

int main() {
    const int size = 10000; // Size of the array (adjust as needed)
    int* array = (int*)malloc(size * sizeof(int));

    generate_random_numbers(array, size);

    int sum = find_sum(array, size);
    int product = find_product(array, size);

    printf("Sum: %d\n", sum);
    printf("Product: %d\n", product);

    free(array);

    return 0;
}
```

```
maakali@kali: ~/profiling
```

```
File Actions Edit View Help
maakali@kali:~/profiling x maakali@kali:~/profiling x
GNU nano 7.2
#include <stdio.h>
#include <stdlib.h>
// A function to generate random numbers and store them in an array
void generate_random_numbers(int* array, int size) {
for (int i = 0; i < size; i++) {
array[i] = rand() % 1000;
}
// A function to find the sum of elements in an array
int find_sum(int* array, int size) {
int sum = 0;
for (int i = 0; i < size; i++) {
sum += array[i];
}
return sum;
}
// A function to find the product of elements in an array
int find_product(int* array, int size) {
int product = 1;
for (int i = 0; i < size; i++) {
product *= array[i];
}
return product;
}
int main() {
const int size = 10000; // Size of the array (adjust as needed)
int* array = (int*)malloc(size * sizeof(int));
generate_random_numbers(array, size);
int sum = find_sum(array, size);
int product = find_product(array, size);
printf("Sum: %d\n", sum);
printf("Product: %d\n", product);
free(array);
return 0;
}
```

File / Applications / Accessories / Text Editor - Kali Linux - /profiling

maakali@kali: ~/profiling

Read 25 lines

```
maakali@kali: ~/profiling
```

```
File Actions Edit View Help
maakali@kali:~/profiling x maakali@kali:~/profiling x
```

```
[maakali@kali:~/profiling]
$ nano p1.c
[maakali@kali:~/profiling]
$ gcc -pg -o p1 p1.c
[maakali@kali:~/profiling]
$ ./p1
Sum: 4969211
Product: 0
[maakali@kali:~/profiling]
$ gprof p1 gmon.out > report.txt
[maakali@kali:~/profiling]
$ nano report.txt
[maakali@kali:~/profiling]
$
```

maakali@kali: ~/profiling

File Actions Edit View Help

GNU nano 7.2 report5.txt

**Flat profile:**

Each sample counts as 0.01 seconds.  
no time accumulated

%	cumulative	self	self	total
time	seconds	seconds	calls	Ts/call Ts/call name
0.00	0.00	0.00	1	0.00 0.00 find_product
0.00	0.00	0.00	1	0.00 0.00 find_sum
0.00	0.00	0.00	1	0.00 0.00 generate_random_numbers

% the percentage of the total running time of the  
time program used by this function.

cumulative a running sum of the number of seconds accounted  
seconds for by this function and those listed above it.

self the number of seconds accounted for by this  
seconds function alone. This is the major sort for this  
listing.

calls the number of times this function was invoked, if  
this function is profiled, else blank.

self ms/call the average number of milliseconds spent in this  
function per call, if this function is profiled,  
else blank.

total ms/call the average number of milliseconds spent in this  
function and its descendants per call, if this  
function is profiled, else blank.

name the name of the function. This is the minor sort  
for this listing. The index shows the location of  
the function in the gprof listing. If the index is  
in parenthesis it shows where it would appear in  
the gprof listing if it were to be printed.

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

maakali@kali: ~/profiling

File Actions Edit View Help

GNU nano 7.2 report5.txt

Copyright (C) 2012-2023 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,  
are permitted in any medium without royalty provided the copyright  
notice and this notice are preserved.

^L Call graph (explanation follows)

granularity: each sample hit covers 4 byte(s) no time propagated

index	% time	self	children	called	name
[1]	0.0	0.00	0.00	1/1	main [9]
					find_product [1]
[2]	0.0	0.00	0.00	1/1	main [9]
					find_sum [2]
[3]	0.0	0.00	0.00	1/1	main [9]
					generate_random_numbers [3]

This table describes the call tree of the program, and was sorted by the total amount of time spent in each function and its children.

Each entry in this table consists of several lines. The line with the index number at the left hand margin lists the current function. The lines above it list the functions that called this function, and the lines below it list the functions this one called.

This line lists:

- index A unique number given to each element of the table.
- Index numbers are sorted numerically.
- The index number is printed next to every function name so it is easier to look up where the function is in the table.

% time This is the percentage of the 'total' time that was spent in this function and its children. Note that due to

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

maakali@kali: ~/profiling

File Actions Edit View Help

GNU nano 7.2 report5.txt

self This is the total amount of time spent in this function.

children This is the total amount of time propagated into this function by its children.

called This is the number of times the function was called. If the function called itself recursively, the number only includes non-recursive calls, and is followed by a '+' and the number of recursive calls.

name The name of the current function. The index number is printed after it. If the function is a member of a cycle, the cycle number is printed between the function's name and the index number.

For the function's parents, the fields have the following meanings:

self This is the amount of time that was propagated directly from the function into this parent.

children This is the amount of time that was propagated from the function's children into this parent.

called This is the number of times this parent called the function '/' the total number of times the function was called. Recursive calls to the function are not included in the number after the '/'.

name This is the name of the parent. The parent's index number is printed after it. If the parent is a member of a cycle, the cycle number is printed between the name and the index number.

If the parents of the function cannot be determined, the word '<spontaneous>' is printed in the 'name' field, and all the other

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

maakali@kali: ~/profiling

File Actions Edit View Help

GNU nano 7.2 report5.txt

For the function's children, the fields have the following meanings:

self This is the amount of time that was propagated directly from the child into the function.

children This is the amount of time that was propagated from the child's children to the function.

called This is the number of times the function called this child `/' the total number of times the child was called. Recursive calls by the child are not listed in the number after the `/'.

name This is the name of the child. The child's index number is printed after it. If the child is a member of a cycle, the cycle number is printed between the name and the index number.

If there are any cycles (circles) in the call graph, there is an entry for the cycle-as-a-whole. This entry shows who called the cycle (as parents) and the members of the cycle (as children.) The `+' recursive calls entry shows the number of function calls that were internal to the cycle, and the calls entry for each member shows, for that member, how many times it was called from other members of the cycle.

^L

Copyright (C) 2012-2023 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification, are permitted in any medium without royalty provided the copyright notice and this notice are preserved.

^L

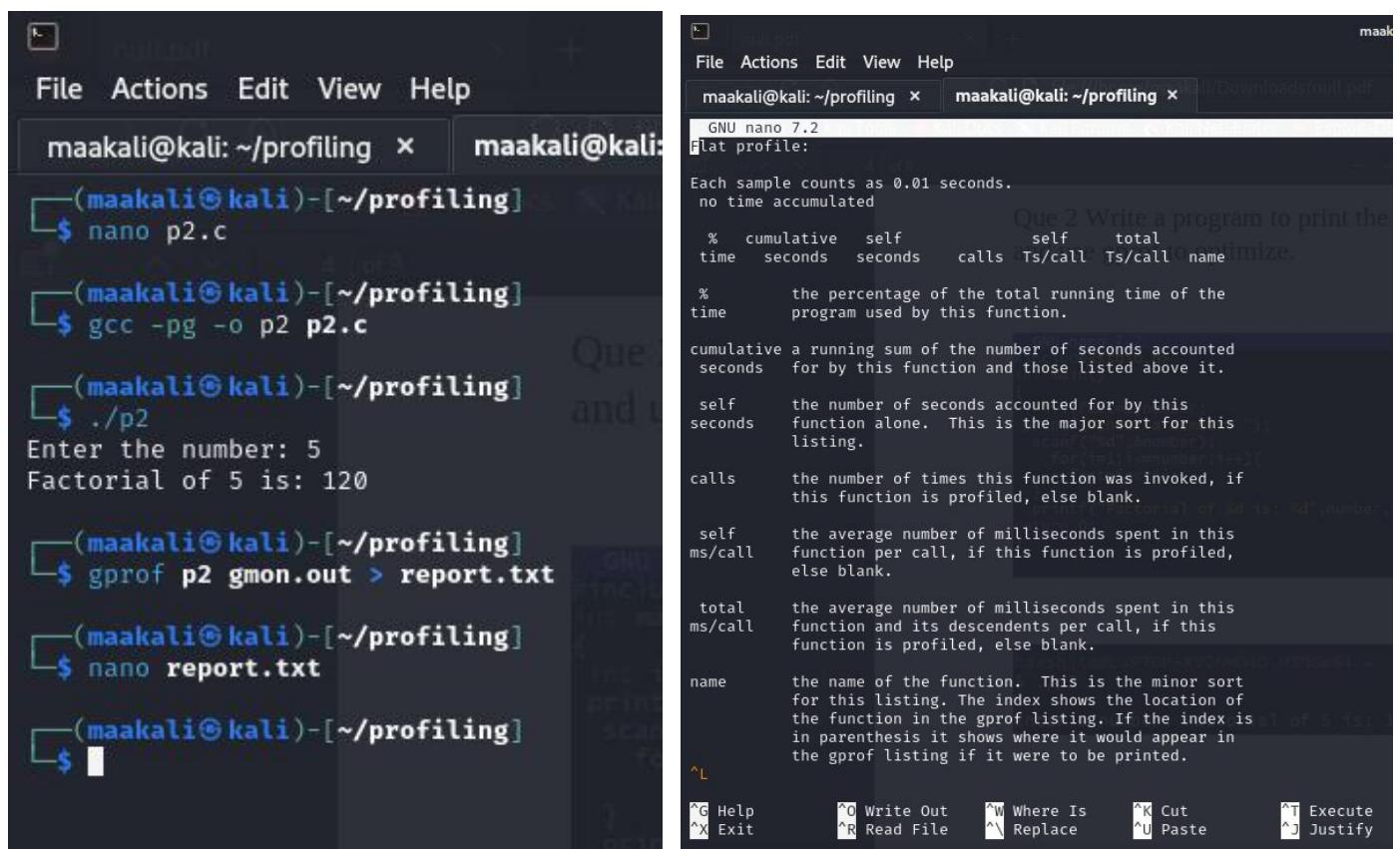
Index by function name

[1] find\_product [2] find\_sum [3] generate\_random\_

Que 2 Write a program to print the factorial of a number which is taken as input and use gprof to optimize.



```
maakali@kali: ~/profiling
File Actions Edit View Help
maakali@kali: ~/profiling × maakali@kali: ~/profiling ×
GNU nano 7.2
#include <stdio.h>
int main(){
    int i,fact=1,number;
    printf("Enter the number: ");
    scanf("%d",&number);
    for(i=1; i≤number; i++){
        fact=fact*i;
    }
    printf("Factorial of %d is: %d",number,fact);
return 0;
}
```



```
File Actions Edit View Help
maakali@kali: ~/profiling × maakali@kali: ~/profiling ×
maakali@kali: ~/profiling [~/profiling]
$ nano p2.c
maakali@kali: ~/profiling [~/profiling]
$ gcc -pg -o p2 p2.c
maakali@kali: ~/profiling [~/profiling]
$ ./p2
Enter the number: 5
Factorial of 5 is: 120
maakali@kali: ~/profiling [~/profiling]
$ gprof p2 gmon.out > report.txt
maakali@kali: ~/profiling [~/profiling]
$ nano report.txt
maakali@kali: ~/profiling [~/profiling]
$
```

Flat profile:

Each sample counts as 0.01 seconds.  
no time accumulated

% time	cumulative seconds	self seconds	calls	Ts/call	Ts/call	name
% time	the percentage of the total running time of the program used by this function.					
cumulative seconds	a running sum of the number of seconds accounted for by this function and those listed above it.					
self seconds	the number of seconds accounted for by this function alone. This is the major sort for this listing.					
calls	the number of times this function was invoked, if this function is profiled, else blank.					
self ms/call	the average number of milliseconds spent in this function per call, if this function is profiled, else blank.					
total ms/call	the average number of milliseconds spent in this function and its descendants per call, if this function is profiled, else blank.					
name	the name of the function. This is the minor sort for this listing. The index shows the location of the function in the gprof listing. If the index is 1 of 5, 15, etc., then the function is the 15th function in the listing.					

^L

^G Help ^O Write Out ^W Where Is ^K Cut  
^X Exit ^R Read File ^\ Replace ^U Paste ^T Execute  
^J Justify

Q3. Write a program for calculator that can perform addition, subtraction, multiplication, division use gprof to optimize.

The image shows two terminal windows side-by-side. The left window displays the source code for a C program named p3.c, which implements a simple calculator. The right window shows the command-line steps to build the program with profiling, run it, and generate a gprof report.

**Left Terminal (Code):**

```
maakali@kali: ~/profiling
File Actions Edit View Help
GNU nano 7.2          p3.c
#include <stdio.h>
int main(){
    char op;
    double first,second;
    printf("Enter an operator (+, -, *, /): ");
    scanf("%c",&op);
    printf("Enter two operands : ");
    scanf("%lf %lf",&first,&second);

    switch (op){
        case '+':
            printf("%.1lf + %.1lf = %.1lf",first, second, first+second);
            break;
        case '-':
            printf("%.1lf - %.1lf = %.1lf",first, second, first-second);
            break;
        case '*':
            printf("%.1lf * %.1lf = %.1lf",first, second, first*second);
            break;
        case '/':
            printf("%.1lf / %.1lf = %.1lf",first, second, first/second);
            break;
        default:
            printf("Error! operator is not correct");
    }
    return 0;
}
```

**Right Terminal (Execution):**

```
maakali@kali: ~
File Actions Edit View Help
(maakali@kali)-[~/profiling]
$ nano p3.c
(maakali@kali)-[~/profiling]
$ gcc -pg -o p3 p3.c
(maakali@kali)-[~/profiling]
$ ./p3
Enter an operator (+, -, *, /): -
Enter two operands : 5 2
5.0 - 2.0 = 3.0
(maakali@kali)-[~/profiling]
$ gprof p3 gmon.out > report.txt
(maakali@kali)-[~/profiling]
$ nano report.txt
```

```
% cumulative self      self      total
time    seconds   seconds   calls Ts/call Ts/call  name
%           the percentage of the total running time of the
time      program used by this function.

cumulative a running sum of the number of seconds accounted
seconds   for by this function and those listed above it.

self      the number of seconds accounted for by this
seconds   function alone. This is the major sort for this
listing.

calls     the number of times this function was invoked, if
this function is profiled, else blank.

self      the average number of milliseconds spent in this
ms/call   function per call, if this function is profiled,
else blank.

total    the average number of milliseconds spent in this
ms/call   function and its descendants per call, if this
function is profiled, else blank.

name      the name of the function. This is the minor sort
for this listing. The index shows the location of
the function in the gprof listing. If the index is
in parenthesis it shows where it would appear in
the gprof listing if it were to be printed.
```

^L  
Copyright (C) 2012-2023 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Que 4 Calculate the area and perimeter of common shapes (circle, rectangle, triangle, square) and use gprof to optimize.

```

File Actions Edit View Help
GNU nano 7.2 p4.c *
#include<stdio.h>
void square(int a){
    printf("Area of Sqaure is %d\n",a*a);
    printf("perimeter of Sqaure is %d\n",4*a);
}
void rectangle(int l,int b){
    printf("Area of Rectangle is %d\n",l*b);
    printf("perimeter of Rectangle is %d\n",2*l+b);
}
void circle(int r){
    printf("Area of circle is %.2f\n",3.14*r*r);
    printf("perimeter of circle is %.2f\n",2*3.14*r);
}
void triangle(int base, int height, int p,int q, int s){
    printf("Area of triangle is %.2f\n",0.5*base*height);
    printf("perimeter of triagnle is %d\n",p+q+s);■
}
int main(){
    int a,l,b,r,base,height,p,q,s;
    printf("Enter sides of square : \n");

    scanf("%d",&a);
    printf("Enter length and breadth of rectangle : \n");
    scanf("%d",&l);
    scanf("%d",&b);
    printf("Enter the radius of circle: \n");
    scanf("%d",&r);
    printf("Enter the base and height of triangle : \n");
    scanf("%d",&base);
    scanf("%d",&height);
    printf("Enter sides of triangle: \n");
    scanf("%d",&p);
    scanf("%d",&q);
    scanf("%d",&s);
    square(a);
    rectangle(l,b);
    circle(r);
    triangle(base,height,p,q,s);
    return 0;
}

```

```

File Actions Edit View Help
└─(maakali㉿kali)-[~/profiling]
└─$ ls
gmon.out p1 p1.c p2 p2.c p3 p3.c report.txt report5.txt report6.txt report7.txt
└─(maakali㉿kali)-[~/profiling]
└─$ nano p4.c
└─(maakali㉿kali)-[~/profiling]
└─$ gcc -pg -o p4 p4.c
└─(maakali㉿kali)-[~/profiling]
└─$ ./p4
Enter sides of square :
5
Enter length and breadth of rectangle :
2
3
Enter the radius of circle:
2
Enter the base and height of triangle :
5
5
Enter sides of triangle:
5
5
5
Area of Sqaure is 25
perimeter of Sqaure is 20
Area of Rectangle is 6
perimeter of Rectangle is 12
Area of circle is 12.56
perimeter of circle is 12.56
Area of triangle is 12.5
perimeter of triagnle is 15
└─(maakali㉿kali)-[~/profiling]
└─$ gprof p4 gmon.out > report8.txt
└─(maakali㉿kali)-[~/profiling]
└─$ nano report8.txt

```

File Actions Edit View Help

GNU nano 7.2 report8.txt

Flat profile:

Each sample counts as 0.01 seconds.  
no time accumulated

% cumulative	self seconds	calls	self Ts/call	total Ts/call	name
0.00	0.00 0.00	1	0.00	0.00	circle
0.00	0.00 0.00	1	0.00	0.00	rectangle
0.00	0.00 0.00	1	0.00	0.00	square
0.00	0.00 0.00	1	0.00	0.00	triangle

% the percentage of the total running time of the  
time program used by this function.

cumulative a running sum of the number of seconds accounted  
seconds for by this function and those listed above it.

self the number of seconds accounted for by this  
seconds function alone. This is the major sort for this  
listing.

calls the number of times this function was invoked, if  
this function is profiled, else blank.

self ms/call the average number of milliseconds spent in this  
function per call, if this function is profiled,  
else blank.

total ms/call the average number of milliseconds spent in this  
function and its descendants per call, if this  
function is profiled, else blank.

name the name of the function. This is the minor sort  
for this listing. The index shows the location of  
the function in the gprof listing. If the index is  
in parenthesis it shows where it would appear in

**^G Help** **^O Write Out** **^W Where Is** **^K Cut** **^T Execute** **^C Location**  
**^X Exit** **^R Read File** **^\\ Replace** **^U Paste** **^J Justify** **^/ Go To Line**

File Actions Edit View Help

GNU nano 7.2 report8.txt

Copyright (C) 2012-2023 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,  
are permitted in any medium without royalty provided the copyright  
notice and this notice are preserved.

**^L** Call graph (explanation follows)

granularity: each sample hit covers 4 byte(s) no time propagated

index	% time	self	children	called	name
[1]	0.0	0.00	0.00	1/1	main [10]
					circle [1]
[2]	0.0	0.00	0.00	1/1	main [10]
					rectangle [2]
[3]	0.0	0.00	0.00	1/1	main [10]
					square [3]
[4]	0.0	0.00	0.00	1/1	main [10]
					triangle [4]

This table describes the call tree of the program, and was sorted by  
the total amount of time spent in each function and its children.

Each entry in this table consists of several lines. The line with the  
index number at the left hand margin lists the current function.  
The lines above it list the functions that called this function,  
and the lines below it list the functions this one called.

This line lists:

- index A unique number given to each element of the table.
- Index numbers are sorted numerically.
- The index number is printed next to every function name so  
it is easier to look up where the function is in the table.

% time This is the percentage of the 'total' time that was spent  
in this function and its children. Note that due to  
different viewpoints, functions excluded by options, etc,

```

File Actions Edit View Help
GNU nano 7.2                                         report8.txt

self      This is the total amount of time spent in this function.

children  This is the total amount of time propagated into this
          function by its children.

called    This is the number of times the function was called.
          If the function called itself recursively, the number
          only includes non-recursive calls, and is followed by
          a '+' and the number of recursive calls.

name      The name of the current function. The index number is
          printed after it. If the function is a member of a
          cycle, the cycle number is printed between the
          function's name and the index number.

For the function's parents, the fields have the following meanings:

self      This is the amount of time that was propagated directly
          from the function into this parent.

children  This is the amount of time that was propagated from
          the function's children into this parent.

called    This is the number of times this parent called the
          function '/' the total number of times the function
          was called. Recursive calls to the function are not
          included in the number after the '/'.

name      This is the name of the parent. The parent's index
          number is printed after it. If the parent is a
          member of a cycle, the cycle number is printed between
          the name and the index number.

If the parents of the function cannot be determined, the word
'<spontaneous>' is printed in the 'name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

```

```

File Actions Edit View Help
GNU nano 7.2                                         report8.txt

If the parents of the function cannot be determined, the word
'<spontaneous>' is printed in the 'name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self      This is the amount of time that was propagated directly
          from the child into the function.

children  This is the amount of time that was propagated from the
          child's children to the function.

called    This is the number of times the function called
          this child '/' the total number of times the child
          was called. Recursive calls by the child are not
          listed in the number after the '/'.

name      This is the name of the child. The child's index
          number is printed after it. If the child is a
          member of a cycle, the cycle number is printed
          between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The '+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.

^L
Copyright (C) 2012-2023 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

^L
Index by function name

[1] circle           [3] square
[2] rectangle        [4] triangle

```



# **Debugging Assignment**

Q1.

```
#include int main() {  
    int a,b;  
    int c=a+b;  
    printf("%d",c); // Output: 5  
    return 0;  
}
```

```
maakali@kali: ~/debugging  
File Actions Edit View Help  
  
└─(maakali㉿kali)-[~/debugging]  
└─$ gcc -g gdb1.c  
  
└─(maakali㉿kali)-[~/debugging]  
└─$ gdb a.out  
GNU gdb (Debian 13.2-1) 13.2  
Copyright (C) 2023 Free Software Foundation, Inc.  
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>  
This is free software: you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
Type "show copying" and "show warranty" for details.  
This GDB was configured as "i686-linux-gnu".  
Type "show configuration" for configuration details.  
For bug reporting instructions, please see:  
<https://www.gnu.org/software/gdb/bugs/>.  
Find the GDB manual and other documentation resources online at:  
<http://www.gnu.org/software/gdb/documentation/>.  
  
For help, type "help".  
Type "apropos word" to search for commands related to "word" ...  
Reading symbols from a.out ...
```

```
maakali@kali: ~/debugging  
File Actions Edit View Help  
  
(gdb) run  
Starting program: /home/maakali/debugging/a.out  
[Thread debugging using libthread_db enabled]  
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".  
1874726306[Inferior 1 (process 21893) exited normally]  
(gdb) break 2  
Breakpoint 1 at 0x4011a5: file gdb1.c, line 5.  
(gdb) r  
Starting program: /home/maakali/debugging/a.out  
[Thread debugging using libthread_db enabled]  
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".  
  
Breakpoint 1, main () at gdb1.c:5  
5      int c=a+b;  
(gdb) p a  
$1 = -1208204288  
(gdb) p b  
$2 = -1212036702  
(gdb) a=2, b=3  
Ambiguous command "a=2, b=3": .  
(gdb) r  
The program being debugged has been started already.  
Start it from the beginning? (y or n) y  
Starting program: /home/maakali/debugging/a.out  
[Thread debugging using libthread_db enabled]  
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".  
  
Breakpoint 1, main () at gdb1.c:5  
5      int c=a+b;
```

Q.2)

```
#include int main() {  
int d=2;  
printf("Enter the value of d:");  
scanf("%d",d);  
printf("The value of d is:%d",d);  
return 0;  
}
```

### Correct code

```
#include <stdio.h>  
int main()  
{  
int d=2;  
printf("Enter the value of d:");  
scanf("%d",&d);  
printf("The value of d is:%d",d);  
return 0;  
}
```

The screenshot shows a terminal window with the following session:

```
maakali@kali: ~/debugging  
File Actions Edit View Help  
[(maakali@kali)-[~/debugging]]  
$ gcc gdb2.c -g -o gdb2  
[(maakali@kali)-[~/debugging]]  
$ gdb ./gdb2  
GNU gdb (Debian 13.2-1) 13.2  
Copyright (C) 2023 Free Software Foundation, Inc.  
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>  
This is free software: you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
Type "show copying" and "show warranty" for details.  
This GDB was configured as "i686-linux-gnu".  
Type "show configuration" for configuration details.  
For bug reporting instructions, please see:  
<https://www.gnu.org/software/gdb/bugs/>.  
Find the GDB manual and other documentation resources online at:  
<http://www.gnu.org/software/gdb/documentation/>.  
  
For help, type "help".  
Type "apropos word" to search for commands related to "word" ...  
Reading symbols from ./gdb2 ...
```

```
maakali@kali: ~/debugging

File Actions Edit View Help
For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from ./gdb2 ...
(gdb) list
1 #include <stdio.h>
2 int main()
3 {
4     int d=2;
5     printf("Enter the value of d:");
6     scanf("%d",d);
7     printf("The value of d is:%d",d);
8     return 0;
9 }
(gdb) break main
Breakpoint 1 at 0x11b6: file gdb2.c, line 4.
(gdb) r
Starting program: /home/maakali/debugging/gdb2
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at gdb2.c:4
4     int d=2;
(gdb) n
5     printf("Enter the value of d:");
(gdb) n
6     scanf("%d",d);
(gdb) n
Enter the value of d:7

Program received signal SIGSEGV, Segmentation fault.
0xb7c61e92 in __vfscanf_internal (s=<optimized out>, format=<optimized out>, argptr=<optimized out>, mode_flags=<optimized out>)
  at ./stdio-common/vfscanf-internal.c:1896
1896  __./stdio-common/vfscanf-internal.c: No such file or directory.
```

Q.3)

```
#include <stdio.h>
#include <stdlib.h>
int factorial(int n);
int main(void) {
    int n = 5;
    int f = factorial(n);
    printf("The factorial of %d is %d.\n", n, f); n = 17;
    f = factorial(n);
    printf("The factorial of %d is %d.\n", n, f); return 0;
}
//A factorial is calculated by n! = n * (n - 1) * (n - 2) * ... * 1 //E.g. 5! = 5 * 4 * 3 * 2 * 1 = 120
int factorial(int n)
{ int f = 1; int i = 1;
while (i <= n) { f = f * i; i++; }
return f;
}
```

### Correct code Q.3

```
#include <stdio.h>
#include <stdlib.h>
long long int factorial(int n);
int main(void) {
int n = 5;
long long int f = factorial(n);
printf("The factorial of %d is %lld.\n", n, f);
n = 17;
f = factorial(n);
printf("The factorial of %d is %lld.\n", n, f);
return 0;
}
//A factorial is calculated by n! = n * (n - 1) * (n - 2) * ... * 1
//E.g. 5! = 5 * 4 * 3 * 2 * 1 = 120
long long int factorial(int n) {
long long int f = 1;
int i = 1;
while (i <= n) {
f = f * i;
i++;
}
return f;
}
```

```
maakali@kali: ~/debugging
File Actions Edit View Help
maakali@kali: ~/debugging x maakali@kali: ~/debugging x

└─(maakali㉿kali)-[~/debugging]
$ gcc gdb3.c -g -o gdb3

└─(maakali㉿kali)-[~/debugging]
$ gdb ./gdb3
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from ./gdb3 ...
```

```
maakali@kali: ~/debugging
File Actions Edit View Help
maakali@kali: ~/debugging x maakali@kali: ~/debugging x

(gdb) run
Starting program: /home/maakali/debugging/gdb3
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".
The factorial of 5 is 120.
The factorial of 17 is -288522240.
[Inferior 1 (process 23612) exited normally]
(gdb) break main
Breakpoint 1 at 0x4011a6: file gdb3.c, line 5.
(gdb) r
Starting program: /home/maakali/debugging/gdb3
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at gdb3.c:5
5      int n = 5;
(gdb) n
6      int f = factorial(n);
(gdb) n
7      printf("The factorial of %d is %d.\n", n, f);
(gdb) n
The factorial of 5 is 120.
8      n = 17;
(gdb) n
9      f = factorial(n);
(gdb) n
10     printf("The factorial of %d is %d.\n", n, f);
(gdb) n
The factorial of 17 is -288522240.
11     return 0;
(gdb) break factorial(n)
Function "factorial(n)" not defined.
```

```
maakali@kali: ~/debugging
File Actions Edit View Help
maakali@kali: ~/debugging x maakali@kali: ~/debugging x

Breakpoint 3 at 0x401225: file gdb3.c, line 16.
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/maakali/debugging/gdb3
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at gdb3.c:5
5     int n = 5;
(gdb) n
6     int f = factorial(n);
(gdb) n

Breakpoint 3, factorial (n=5) at gdb3.c:16
16    int f = 1;
(gdb) n
17    int i = 1;
(gdb) n
18    while (i <= n) {
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) n
Program not restarted.
(gdb) c
Continuing.
The factorial of 5 is 120.

Breakpoint 3, factorial (n=17) at gdb3.c:16
16    int f = 1;
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) n
Program not restarted.
(gdb) c
Continuing.
The factorial of 17 is -288522240.
[Inferior 1 (process 24880) exited normally]
```

Q.4)

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

```

int main(void) {
int arr[2];
arr[3] = 10; // Accessing out of bound
return (0);
}

```

Correct code for Q.4):-

```

#include <stdio.h>
int main(void) {
int arr[4];
arr[3] = 10;
return (0);
}

```

The terminal window shows the following session:

```

maakali@kali: ~/debugging
File Actions Edit View Help

[(maakali@kali)-[~/debugging]
$ gcc gdb4.c -g -o gdb4

[(maakali@kali)-[~/debugging]
$ gdb ./gdb4
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from ./gdb4 ...

maakali@kali: ~/debugging
File Actions Edit View Help

Reading symbols from ./gdb4 ...
(gdb) list
1 #include <stdio.h>
2 int main(void)
3 {
4     int arr[2];
5     arr[3] = 10; // Accessing out of bound
6     return (0);
7 }
8
9
(gdb) break main
Breakpoint 1 at 0x1189: file gdb4.c, line 5.
(gdb) r
Starting program: /home/maakali/debugging/gdb4
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at gdb4.c:5
5     arr[3] = 10; // Accessing out of bound
(gdb) print arr[3]
$1 = -1212008507
(gdb) n
6     return (0);
(gdb) p arr[3]
$2 = 10
(gdb) continue
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0x0000000a in ?? ()
(gdb) quit
A debugging session is active.

Inferior 1 [process 1293] will be killed.

Quit anyway? (y or n) y

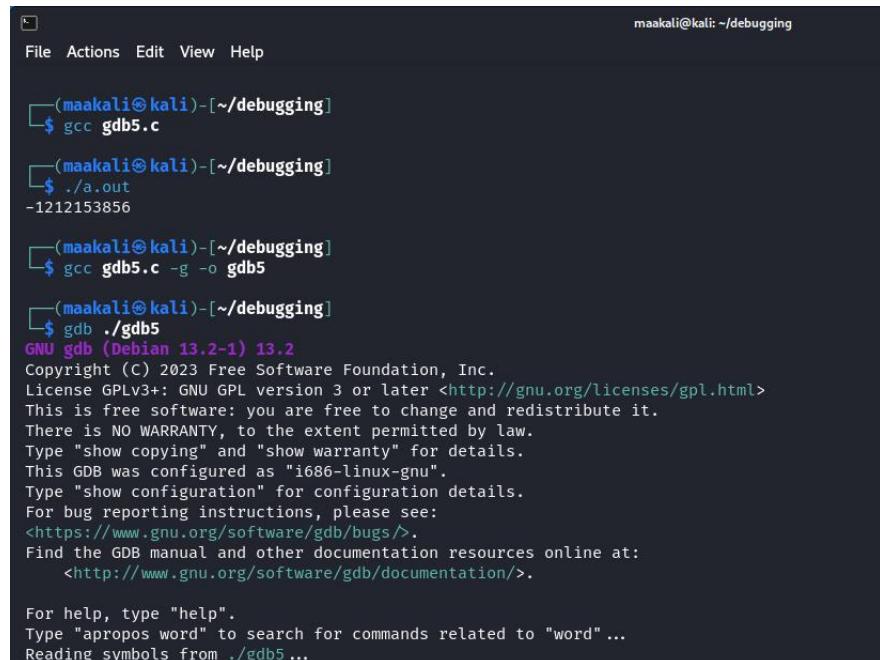
```

## Q.5)

```
#include <stdio.h>
int main(){
    int *p;
    printf("%d",*p);
    return 0;
}
```

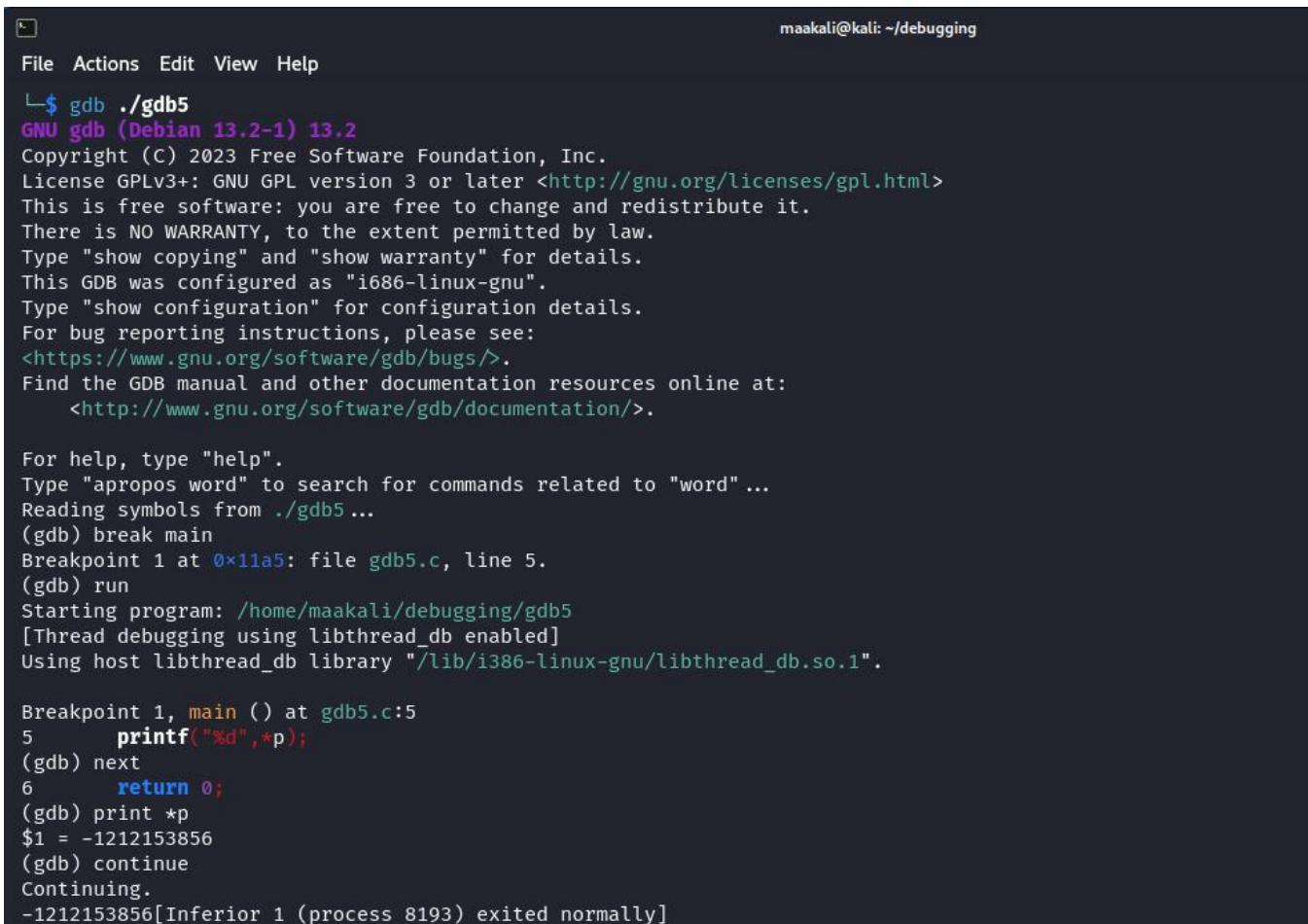
### Correct code Q.5):-

```
#include <stdio.h>
int main()
{
    int *p;
    int var = 15;
    p = &var;
    printf("%d",*p);
    return 0;
}
```



```
maakali@kali: ~/debugging
File Actions Edit View Help
└─(maakali㉿kali)-[~/debugging]
  └─$ gcc gdb5.c
  └─(maakali㉿kali)-[~/debugging]
  └─$ ./a.out
  -1212153856
  └─(maakali㉿kali)-[~/debugging]
  └─$ gcc gdb5.c -g -o gdb5
  └─(maakali㉿kali)-[~/debugging]
  └─$ gdb ./gdb5
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from ./gdb5 ...
```



```
maakali@kali: ~/debugging
File Actions Edit View Help
└─$ gdb ./gdb5
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from ./gdb5 ...
(gdb) break main
Breakpoint 1 at 0x11a5: file gdb5.c, line 5.
(gdb) run
Starting program: /home/maakali/debugging/gdb5
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at gdb5.c:5
5      printf("%d",*p);
(gdb) next
6      return 0;
(gdb) print *p
$1 = -1212153856
(gdb) continue
Continuing.
-1212153856[Inferior 1 (process 8193) exited normally]
```

# **Latex Assignment**

## **Latex Source Code**

```

\documentclass[10pt]{article}
\usepackage{microtype}
\usepackage{graphicx} % Required for inserting images
\usepackage{blindtext}
\usepackage{geometry}
\usepackage{multirow}
\usepackage{enumerate}
\usepackage{tabto}
\usepackage{inputenc}
\usepackage{enumitem,amssymb}
\newlist{todolist}{itemize}{2}
\setlist[todolist]{label=$\square$}
\geometry{
a4paper,
total={170mm,257mm},
left=20mm,
top=10mm,
}
\begin{document}
\begin{flushright}JEE (Advanced) 2022
\vspace{1cm}
Mohan
Manjhi\end{flushright}
$ $\textbf{-----} \\
-----} $ $
\begin{center}
\fbox{\parbox{6.5in}{%
\begin{center}
\includegraphics[width=0.25\linewidth]{iitB.png}
\end{center}
\hspace*{4ex}Name : Mohan Manjhi \tab \hspace*{10ex}Max Marks : 70\\ \\
\hspace*{4ex}Branch : C.S.E \tab \hspace*{10ex}Min Marks : 22\\ \\
\hspace*{4ex}Subject : Latex
Assignment\tab \hspace*{10ex}Time : 03:00 HRS \\ \\
}}}
```

\hspace\*{4ex}Signature of the Examiner with date \tab \hspace\*{10ex}Signature of the Scrutinizer  
with date\\ \\

\hspace\*{4ex}-----\tab \hspace\*{12ex}-----  
-----\\ \\

}

\end{center}

READ THE INSTRUCTIONS CAREFULLY \\

GENERAL \\

\begin{itemize}

\item This sealed booklet is your Question Paper. Do not break the seal till you are told to do so.

\end{itemize}

\begin{itemize}

\item Immediately after breaking the seal of the booklet, verify that the booklet contains 23 pages and that all the 16 questions along with the options are legible.

\end{itemize}

\begin{itemize}

\item Report immediately about any missing or torn sheet in this booklet and ask for a replacement of the booklet from the invigilator. No replacement will be allowed after 15 minutes of the starting of the examination.

\end{itemize}

\begin{itemize}

\item Write your Name, Roll Number, JEE (Advanced) 2022 Rank and other details in the space provided and nowhere else. No distinctive / identification mark of any type is to be put anywhere else in this booklet.

\end{itemize}

\begin{itemize}

\item Answers are to be written only in the booklet and within the space provided beside / below each Question and nowhere else. Answers written in nondesignated place will not be evaluated.

\end{itemize}

\begin{itemize}

\item Blank spaces are provided within this booklet for rough work.

\end{itemize}

\begin{itemize}

\item Do not deface this booklet or detach or mutilate any sheet from the booklet. Such acts lead to disqualification.

\end{itemize}

```
\begin{itemize}
```

\item All the questions are compulsory.

\end{itemize}

\newpage

\begin{flushright}JEE (Advanced) 2022\end{flushright}

Mohan

Manjhi\end{flushright}

\$\$\textbf{--}

-----}\$\$\$

\begin{center}

\fbox{\parbox{6.5in}{\centering

\textbf{SECTION A: Architectural Awareness (Maximum Marks: 60)}\\

Th

and 4 carry \t

\end{center}

%\begin{center}\textbf{SECTION A: Architectural Awareness (Maximum Marks: 60)}\end{center}  
% \\ %This section contains EQUB questions. Question 1 carries 20 marks. Questions 2, 3 \\ %and

4 carry 10 marks each. There is NO negative marking.

%\end{center}

Q.1. This section contains **15** multiple choice questions. Each question has four options, out of which

ONLY\

\hspace\*{5ex}ONF is correct. Mark the correct option with a tick.

\begin{flushright}

**\textbf{(5 x 2 = 10 Marks)}**

\end{flushright}

(i) Which of the following bridges is laid over a sea? \\ \\

\hspace\*{4ex}A. Godavari Bridge, Andhra Pradesh \\ \\

\hspace\*{4ex}B. Pamban Bridge, Tamilnadu \\ \\

\hspace\*{4ex}C. Howrah Bridge, West Bengal \\\

\hspace\*{4ex}D. Mahatma Gandhi Setu, Bihar \\

%

\\ \\

(ii) In 2021, -----annual festivity got inscribed on the representative list of Intangible Cultural \\

\hspace\*{4ex}Heritage of Humanity by UNESCO.

\\ % \\

\hspace\*{4ex}A. Makara Sankranti \tab B. Ganesh Chaturthi \\\ \\

\hspace\*{4ex}C. Durga Puja \tab D. Chhath Puja

\\ \\ % \\

(iii) Statue of which famous personality is called 'Statue of Unity'?

\\ \\ % \\

\hspace\*{4ex}A. Mahatma Gandhi \tab B. Dr. B. R. Ambedkar

\\ \\ \\

\hspace\*{4ex}C. Sardar Vallabhbhai Patel \tab D. Indira Gandhi

\\ \\ % \\

(iv) Which of the following historic gateways is sharing an edge with a water body? \\

\\ % \\

\hspace\*{4ex}A. Charminar, Hyderabad \tab B. India Gate, New Delhi \\\ \\

\hspace\*{4ex}C. Rumi Darwaza, Lucknow \tab D. Gateway of India, Mumbai

\\ \\

%

\\ \\

(v) Which architect has designed the building 'Church of Light', located in Japan?

\\ \\ % \\

\hspace\*{4ex}A. Moshe Safdie \tab B. Fumihiko Maki

\\ \\ \\

\hspace\*{4ex}C. Zaha Hadid \tab D. Tadao Ando

\\ \\

\$\$\textbf{-----}

-----} \$\$

%

\begin{flushright}JEE (Advanced) 2022\end{flushright}

Mohan  
Manjhi\end{flushright}  
\$\\$\\textbf{-----}  
-----}\$\$

**Q.2 Match the famous architects in column I with their renowned projects in column II.**

\hspace{10ex}\textbf{( Mark : 05)}\\

%

11

\hspace\*{4ex}\textbf{Column I} \tab \textbf{Column II}

11

\hspace\*{4ex}(A) Charles Correa \tab (P) IIM Bangalore

11

\hspace\*{4ex}(B) B V Doshi \tab (Q) Capitol Complex, Chandigarh

11

\hspace\*{4ex}(C) Bimal Pate \tab (R) Central Vista, New Delhi

11

\hspace\*{4ex}(D) Raj Rewal \tab (S) Dakshina Chitra, Chennai

11

\hspace\*{4ex}(E) Le Corbusier \tab (T) Bharat Bhawan, Bhopal

11

\hspace\*{4ex} \tab (U) Parliament Library, New Delhi

11

%

Q.3 Match the material in Column I with the craft in Column II. \hspace{28ex}\textbf{( Mark : 05)}\|

%

11

\hspace\*{4ex}\textbf{Column I} \tab \textbf{Column II}

11

\hspace\*{4ex}(A) Stone \tab (P) Origami

11

\hspace\*{4ex}(B) Paper \tab (Q) Embroidery

11

**Q.4** Refer to the following sketch of St. James' Church, Delhi. Match the numbers mentioned in Column I

(highlighted in the sketch) with corresponding building elements in Column II.\hspace{10ex}\textbf{( Mark : 05)}

```
\includegraphics[width=1\linewidth]{q3.jpg}
\\\\\\
%
\\
\hspace*{4ex}\textbf{Column I} \tab \textbf{Column II}
\\\\\\
\hspace*{4ex}(A) i \tab (P) Column
\\\\\\
\hspace*{4ex}(B) ii \tab (Q) Pediment
\\\\\\
\hspace*{4ex}(C) iii \tab (R) Plinth
```

graphical representation shown in figure 2, indicate? \tab \hspace{40ex}\textbf{( Mark : 07)}\\ \\ % \\

\hspace\*{10ex}\includegraphics[width=0.40\linewidth]{q6\_fig1.jpg}

\hspace\*{4ex}\includegraphics[width=0.36\linewidth]{q6\_fig2.jpg} \\ \\

\hspace\*{25ex} Figure 1 \tab \hspace\*{15ex} Figure 2

11

%

\hphantom{A. Thermocol} \tab B. Concrete

11

\hspace\*{4ex}C. Steel \tab D.Brick

11

Q.7 The least value of  $\alpha\epsilon$  for which  $4\alpha x^2 + 1/x \geq 1$ , for all  $x > 0$ , is **20** (Mark : 07) \\

11

$\$\\hspace*{4ex}A.\\hspace*{4ex} \\frac{1}{64} \$\\backslash \\backslash$

$\$\\hspace*{4ex}B.\\hspace*{4ex}\\frac{1}{32}\\$\\|\\|$

$\$\\hspace*{4ex}C.\\hspace*{4ex} \\frac{1}{27} \$\\backslash \\backslash$

$\$\\hspace*{4ex}D.\\hspace*{4ex} \\frac{1}{25} \$\\backslash$

\newpage

\begin{flushright}JEE (Advanced) 2022\end{flushright}

Mohan

Manjhi\end{flushright}

\$\$\textbf{--}

-----}\$\$\$

Q.8 A system goes from A to B via two processes I and II as shown in figure. If  $\Delta U_1$  and  $\Delta U_2$  are the changes

in internal energies in the processes I and II respectively, then \hspace{30ex}\textbf{( Mark : 07)} \\

\$\$ \backslash centering

\includegraphics[width=0.25\linewidth]{Q8.jpg} \$\$

$\$\\hspace*{4ex}A.\\hspace*{4ex} \backslash Delta U_{II} > \backslash Delta U_I \$\\ \\$

$\$\\hspace*{4ex}B.\\hspace*{4ex} \Delta U_{II} < \Delta U_I \$\\backslash \\backslash$

$\$\\hspace*{4ex}C.\\hspace*{4ex} \backslash Delta U\_I = \backslash Delta U_{II} \$\\backslash \\backslash$

\hspace\*{4ex}D.\hspace\*{4ex} Relation between  $\Delta U_I$  and  $\Delta U_{II}$  can not be determined \\

%

\\

Q.9 Consumers were polled about their favourite ice cream flavours in a survey. Draw a bar graph for

the following data: \tab \hspace{42ex}\textbf{( Mark : 07)}\\

\begin{tabular}{|c|c|}

\hline

Flavour of Icecream & Frequency \\

\hline

Vanilla & 1 \\

\hline

Strawberry & 5 \\

\hline

Chocolate & 12 \\

\hline

Mint Chocolate & 3 \\

\hline

Others & 6 \\

\hline

\end{tabular} \\

\\ \\

Q.10 Examine the graph below carefully and answer the following questions. The graph depicts the results of a school's students. \tab \hspace{42ex}\textbf{( Mark : 10)}

\includegraphics[width=0.85\linewidth]{q10.jpg} \\

%

\\

\hspace\*{4ex}A. Which year has the smallest difference between the number of kids who passed and those who failed? \\ \\

\hspace\*{4ex}B. In the last five years, what was the average number of kids who failed in school? \\

\\

\hspace\*{4ex}C. How many times have the same number of kids failed? \\ \\

\end{document}

**\* ASSIGNMENT \***

**PROGRAMMING INFRASTRUCTURE**

1. Push below code in git hub and then clone it from github and after correcting the code again push it into any other branch.(in vscode)

```
def prit(n, isPrime):
    isPrime[0] = isPrime[1] = False
    for i in range(2,n):
        isPrime[i] = True
    for p in range(2,n+1):
        if (p*p<=n and isPrime[p] == True):
            for i in range(p*p,n+1,p):
                isPrime[i] = False
            p += 1
def superPrimes(n):
    isPrime = [1 for i in range(n+1)]
    prit(n, isPrime)
    primes = [0 for i in range(2,n+1)]
    j = 0
    for p in range(2,n+1):
        if(isPrime[p]):
            primes[j] = p
            j += 1
        for k in range(j):
            if(isPrime[k+1]):
                print (primes[k],end=" ")
    n = 241
    print ("\nSuper-Primes less than or equal to ", n, " are :")
    superPrimes(n)
```

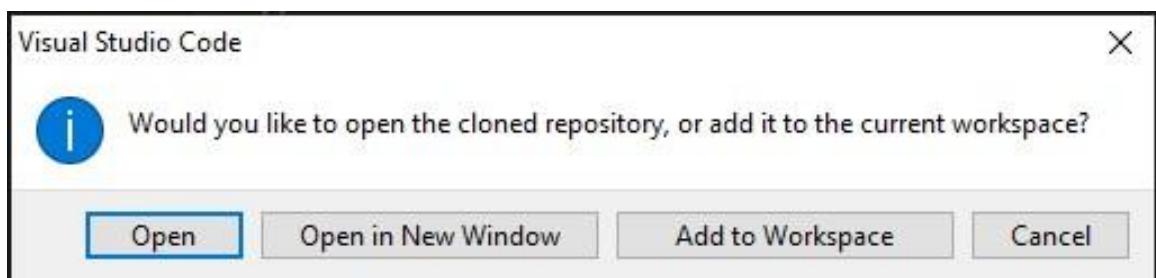
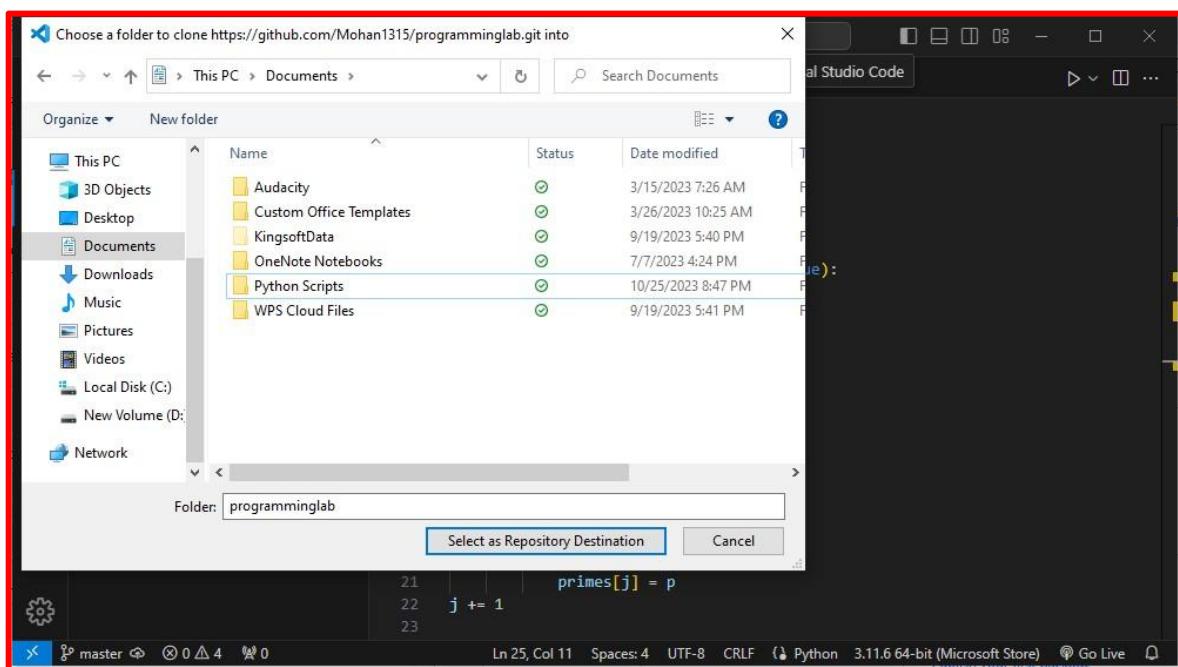
A screenshot of a GitHub repository page. At the top, it shows the branch 'master' with 4 branches and 0 tags. Below this, a message says 'Your master branch isn't protected' with a link to 'Protect this branch'. A list of recent commits by user 'Mohan1315' is shown, all dated '1 minute ago' and titled 'delete some file'. The commits include '.vscode', 'Shoping Cart project', 'movie ticket project', 'Demo.zip', 'ProgrammingLab.zip', and 'program1.py'.

A screenshot of the VS Code interface with a red box highlighting the 'SOURCE CONTROL' tab in the left sidebar. The 'Clone' option is selected in the dropdown menu under 'Clone'.

A screenshot of the VS Code interface with a red box highlighting the code editor. The URL 'https://github.com/Mohan1315/programminglab.git' is entered in the address bar. The code editor displays a Python script named 'program1.py' containing the following code:

```
1  user print(n, isPrime):
2  isPrime[0] = isPrime[1] = False
3  for i in range(2,n):
4      isPrime[i] = True
5      for p in range(2,n+1):
6          if (p*p<=n and isPrime[p] == True):
7              for i in range(p*p,n+1,p):
8                  isPrime[i] = False
9          p += 1
10     def superPrimes(n):
11         isPrime = [1 for i in range(n+1)]
12         print(n, isPrime)
13         primes = [0 for i in range(2,n+1)]
14         j = 0
15         for p in range(2,n+1):
16             if(isPrime[p]):
17                 primes[j] = p
18                 j += 1
19             for k in range(j):
20                 if(isPrime[k+1]):
21                     print(primes[k], end=" ")
22         n = 241
23         print("\nSuper-Primes less than or equal to ", n, " are :")
```

The status bar at the bottom shows 'Ln 25, Col 1 (525 selected) Spaces: 4 UTF-8 Python 3.11.6 64-bit (Microsoft Store)'.



A screenshot of Visual Studio Code showing the cloned repository "PROGRAMMINGLAB". The "program1.py" file is open in the editor. The code defines two functions: `prit(n, isPrime)` and `superPrimes(n)`. The `prit` function initializes a list `isPrime` with `isPrime[0] = isPrime[1] = False` and then iterates from 2 to `n` to set `isPrime[i] = True`. It then iterates from 2 to `n+1` and for each prime `p`, it iterates from `p\*p` to `n+1` and sets `isPrime[i] = False` for all multiples of `p`. The `superPrimes` function creates a list of primes from 2 to `n+1` and then iterates from 2 to `n+1` to update the `primes` list based on the `isPrime` list. The status bar at the bottom shows "Ln 1, Col 1" and "Activating Extensions...".

```
def prit(n, isPrime):
    isPrime[0] = isPrime[1] = False
    for i in range(2,n):
        isPrime[i] = True

    for p in range(2,n+1):
        if (p*p<=n and isPrime[p] == True):
            for i in range(p*p,n+1,p):
                isPrime[i] = False
            p += 1

def superPrimes(n):
    isPrime = [1 for i in range(n+1)]
    prit(n, isPrime)

    primes = [0 for i in range(2,n+1)]
    j = 0

    for p in range(2,n+1):
        if(isPrime[p]):
            primes[j] = p
            j += 1
```

2. Compile the given code and check whether output of the following program is correct or not, if not debug it using vscode and correct the code accordingly.

```
def prit(n, isPrime):
    isPrime[0] = isPrime[1] = False
    for i in range(2,n):
        isPrime[i] = True
    for p in range(2,n+1):
        if (p*p<=n and isPrime[p] == True):
            for i in range(p*p,n+1,p):
                isPrime[i] = False
            p += 1
    def superPrimes(n):
        isPrime = [1 for i in range(n+1)]
        prit(n, isPrime)
        primes = [0 for i in range(2,n+1)]
        j = 0
        for p in range(2,n+1):
            if(isPrime[p]):
                primes[j] = p
                j += 1
            for k in range(j):
                if(isPrime[k+1]):
                    print (primes[k],end=" ")
        n = 241
        print ("\nSuper-Primes less than or equal to ", n, " are :")
        superPrimes(n)
```

```
if (p*p<=n and isPrime[p] == True):
    for i in range(p*p,n+1,p):
        isPrime[i] = False
    p += 1

def superPrimes(n):
    isPrime = [1 for i in range(n+1)]
    print(n, isPrime)

primes = [0 for i in range(2,D n+1)]
```

Exception has occurred: NameError  
name 'n' is not defined  
File "C:\Users\mohan\OneDrive\Documents\programminglab\programminglab\program1.py", line 16, in <module>  
primes = [0 for i in range(2,n+1)]  
^  
NameError: name 'n' is not defined

CALL STACK  
<module> program1.py (16:0)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
PS C:\Users\mohan\OneDrive\Documents\programmingLab\programminglab> & 'C:\Users\mohan\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\mohan\.vscode\extensions\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter/../debugpy\launcher' '65222' '--' 'C:\Users\mohan\OneDrive\Documents\programmingLab\programminglab\program1.py'

```
isPrime[i] = True

for p in range(2,n+1):
    if (p*p<=n and isPrime[p] == True):
        for i in range(p*p,n+1,p):
            isPrime[i] = False
        p += 1

def superPrimes(n):
    isPrime = [1 for i in range(n+1)]
    print(n, isPrime)
    primes = [n for i in range(2,n+1)]
    j = 0
    for p in range(2,n+1):
        if(isPrime[p]):
            primes[j] = p
            j += 1
    for k in range(j):
        if(isPrime[k+1]):
            print (primes[k],end=" ")

n = 241
print ("\nSuper-Primes less than or equal to ", n, " are :")
superPrimes(n)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mohan\OneDrive\Documents\programmingLab\programminglab> python -u "C:\Users\mohan\OneDrive\Documents\programmingLab\programminglab\program1.py"

Super-Primes less than or equal to 241 are :  
3 5 11 17 31 41 59 67 83 109 127 157 179 191 211 241

3. Refactoring below code by using pylance in vscode.

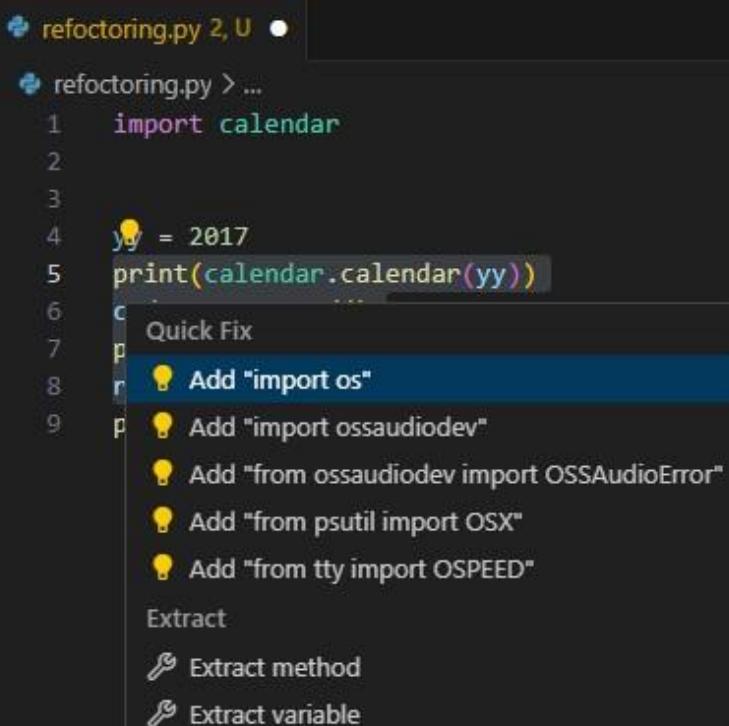
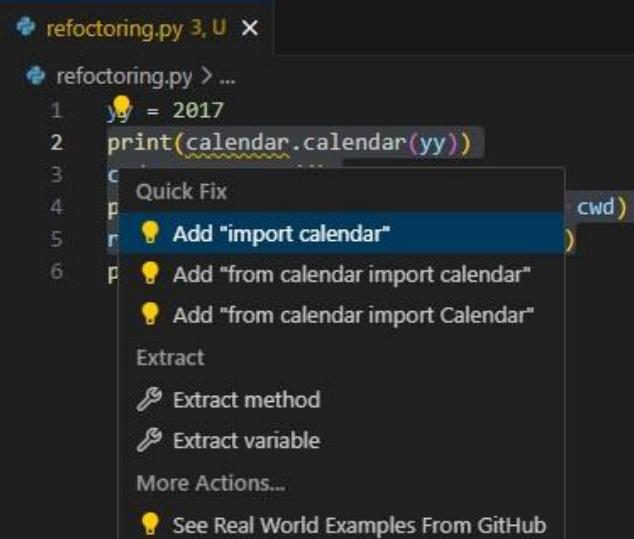
```
yy = 2017
print(calendar.calendar(yy))
cwd = os.getcwd()
print("Current working directory:", cwd)
result = os.path.exists("file_name")
print(result)
```

The screenshot shows the Visual Studio Code interface. The top bar has a search field containing "Infrastructure Assignment". Below it is a tab bar with "refactoring.py 3, U X" selected. The main editor area contains the following Python code:

```
yy = 2017
print(calendar.calendar(yy))
cwd = os.getcwd()
print("Current working directory:", cwd)
result = os.path.exists("file_name")
print(result)
```

The word "calendar" is underlined with a red squiggle, indicating a potential error. The bottom right corner of the editor has a red border. The bottom of the screen shows the terminal window with the following output:

```
llege\Programming Practice\Infrastructure Assignment\refactoring.py"
Traceback (most recent call last):
  File "d:\SGSITS College\Programming Practice\Infrastructure Assignment\refactoring.py", line 2, in <module>
    print(calendar.calendar(yy))
           ^
NameError: name 'calendar' is not defined
PS D:\SGSITS College\Programming Practice\Infrastructure Assignment>
```



The screenshot shows a code editor window titled "Infrastructure Assignment". The left sidebar contains icons for file operations, search, and other tools. The main area displays a Python script named "refactoring.py". The code imports "calendar" and "os", prints a calendar for the year 2017, gets the current working directory, and checks if a file named "file\_name" exists. Below the code, three separate calendar grids are shown for January, February, and March 2017. The terminal tab at the bottom shows the output of running the script in a PowerShell environment, indicating the current working directory and the result of the file check.

```
import calendar
import os
yy = 2017
print(calendar.calendar(yy))
cwd = os.getcwd()
print("Current working directory:", cwd)
result = os.path.exists("file_name")
print(result)
```

Mo	Tu	We	Th	Sa	Su
				1	
2	3	4	5	6	7
9	10	11	12	13	14
16	17	18	19	20	21
23	24	25	26	27	28
30	31				

Mo	Tu	We	Th	Sa	Su
			1	2	3
6	7	8	9	10	11
13	14	15	16	17	18
20	21	22	23	24	25
27	28	29	30		

Mo	Tu	We	Th	Sa	Su
			1	2	3
4	5	6	7	8	9
11	12	13	14	15	16
18	19	20	21	22	23
25	26	27	28	29	30
31					

```
Current working directory: D:\SGSITS College\Programming Practice\Infrastructure Assignment
False
PS D:\SGSITS College\Programming Practice\Infrastructure Assignment>
```

4. Compile the given code and check whether output of the following program is correct or not, if not debug it using any other editor and correct the code accordingly.

```
color = ['red', 'blue', 'green', 'pink']
print(color[len(color)])
```

The screenshot shows the PyCharm IDE interface. The top bar displays the project name "IA Infrastructure Assignment" and the branch "master". The code editor contains the following Python script:

```
color = ['red', 'blue', 'green', 'pink']
print(color[len(color)])
```

The line `print(color[len(color)])` is highlighted in red, indicating a syntax error. The status bar at the bottom right shows the file path "D:\SGSITS College\Programming Practice\Infrastructure Assignment\program4.py", the time "2:25", and the Python version "Python 3.11".

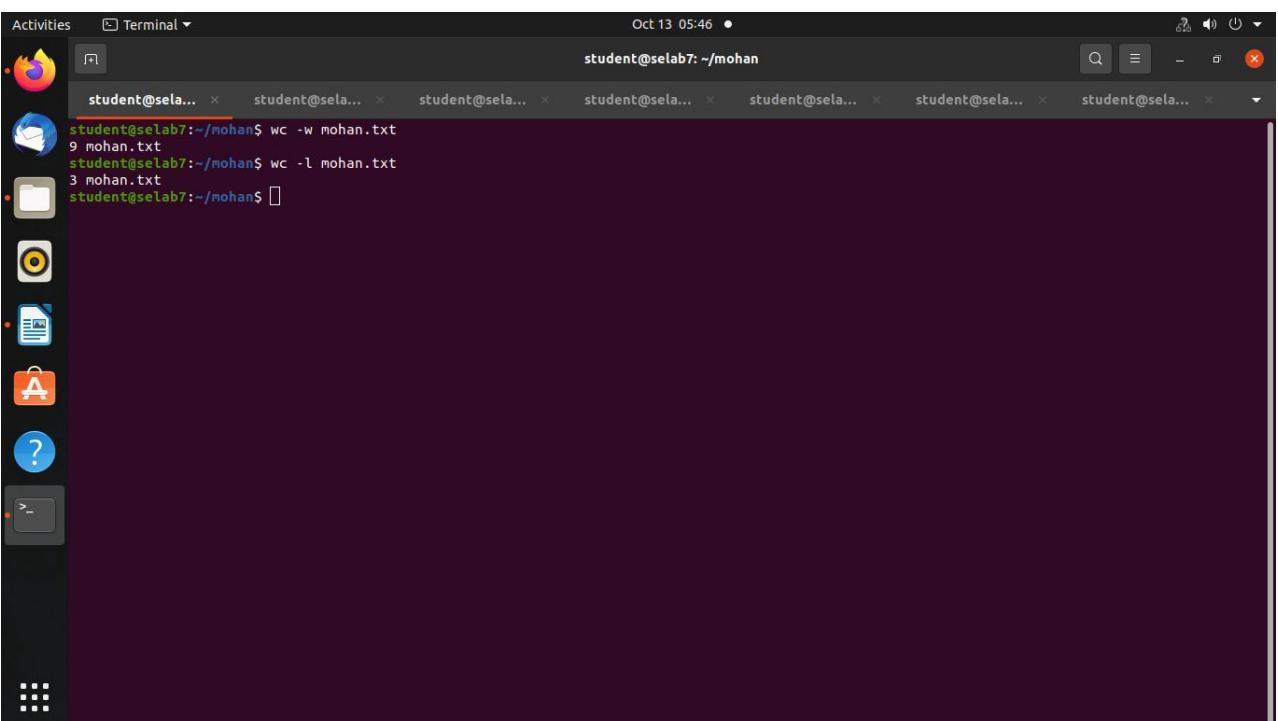
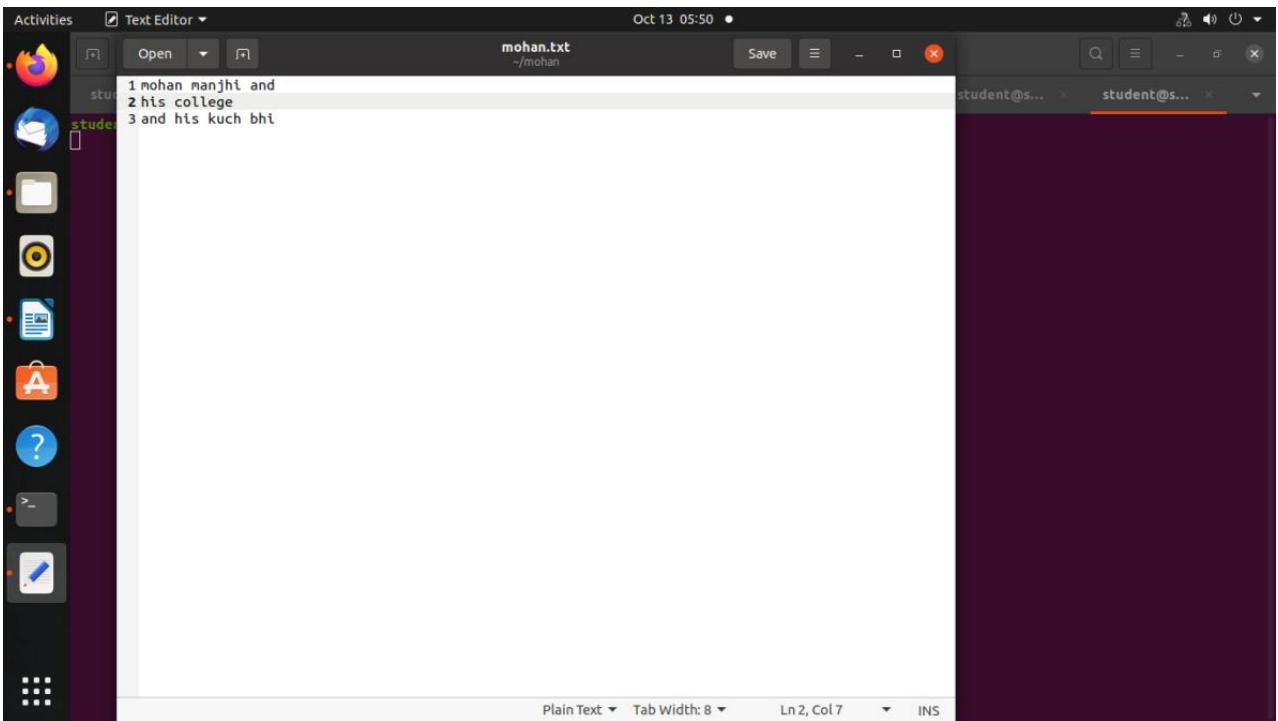
The screenshot shows the PyCharm IDE interface after the code has been modified. The code editor now contains:

```
color = ['red', 'blue', 'green', 'pink']
print(len(color))
```

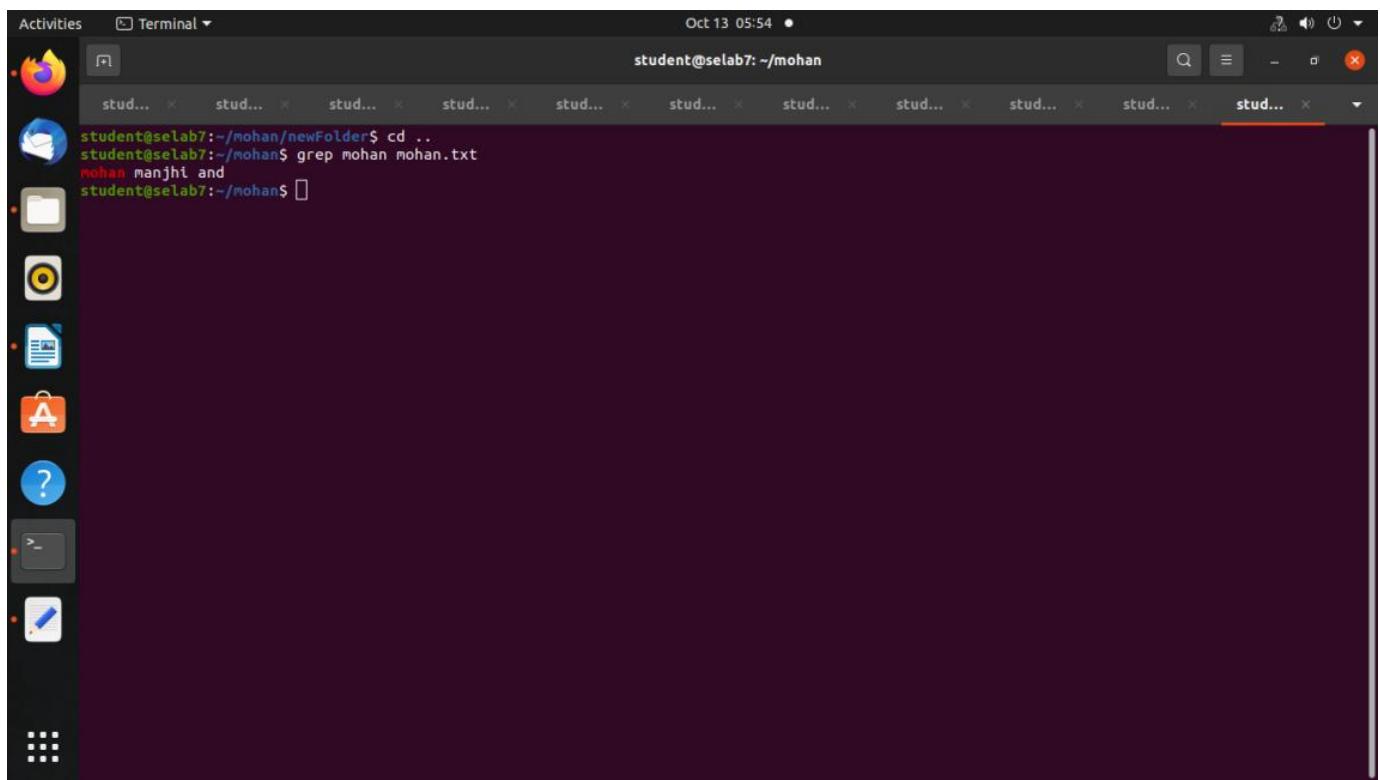
The status bar at the bottom right shows the file path "C:\Users\mohan\AppData\Local\Microsoft\WindowsApps\python3.11.exe", the time "3:1", and the Python version "Python 3.11". The status bar also indicates "Process finished with exit code 0", confirming successful execution.

# **Linux Assignment**

Que 1 Which command is used to count words and line in a text file attach a screenshot which displaying the word and line count.

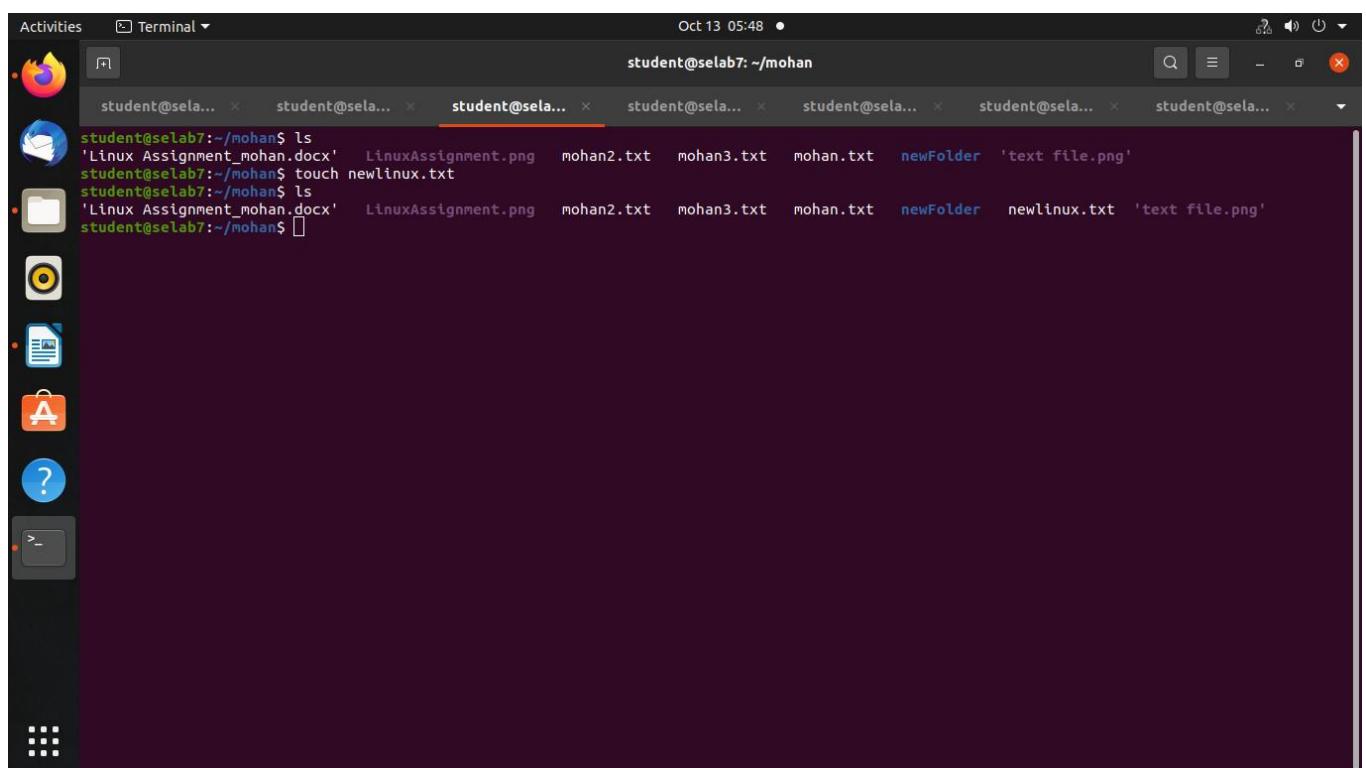


Que 2 Which command is used to search for specific text within a file includes a screenshot showing the result of the search.



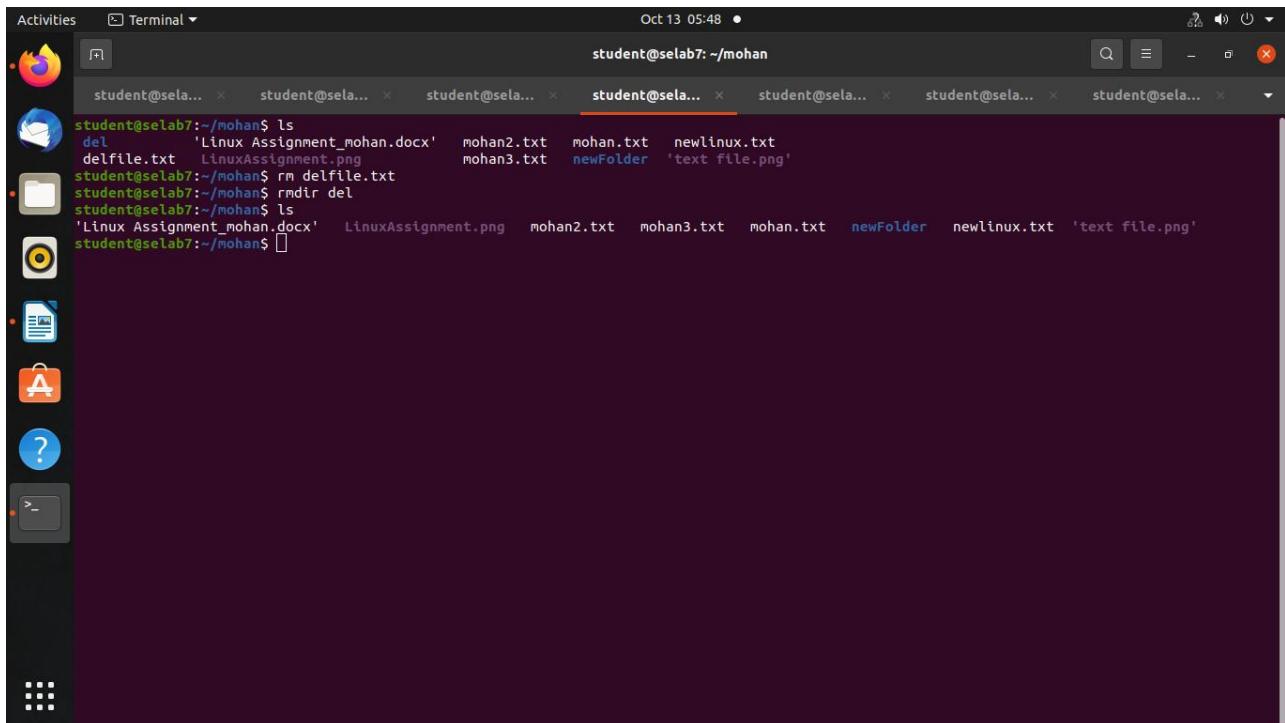
A screenshot of a Linux desktop environment showing a terminal window. The terminal title is "student@selab7: ~/mohan". The command entered was "grep mohan mohan.txt", and the output was "mohan manjhi and". The terminal is part of a docked application bar with other icons like a browser, file manager, and terminal.

Que 3 Create a new empty file using command and capture a screenshot showing the successful creation of a file.



A screenshot of a Linux desktop environment showing a terminal window. The terminal title is "student@selab7: ~/mohan". The commands entered were "ls", "touch newLinux.txt", and "ls" again. The output shows the creation of a new file "newLinux.txt" in the directory. The terminal is part of a docked application bar with other icons like a browser, file manager, and terminal.

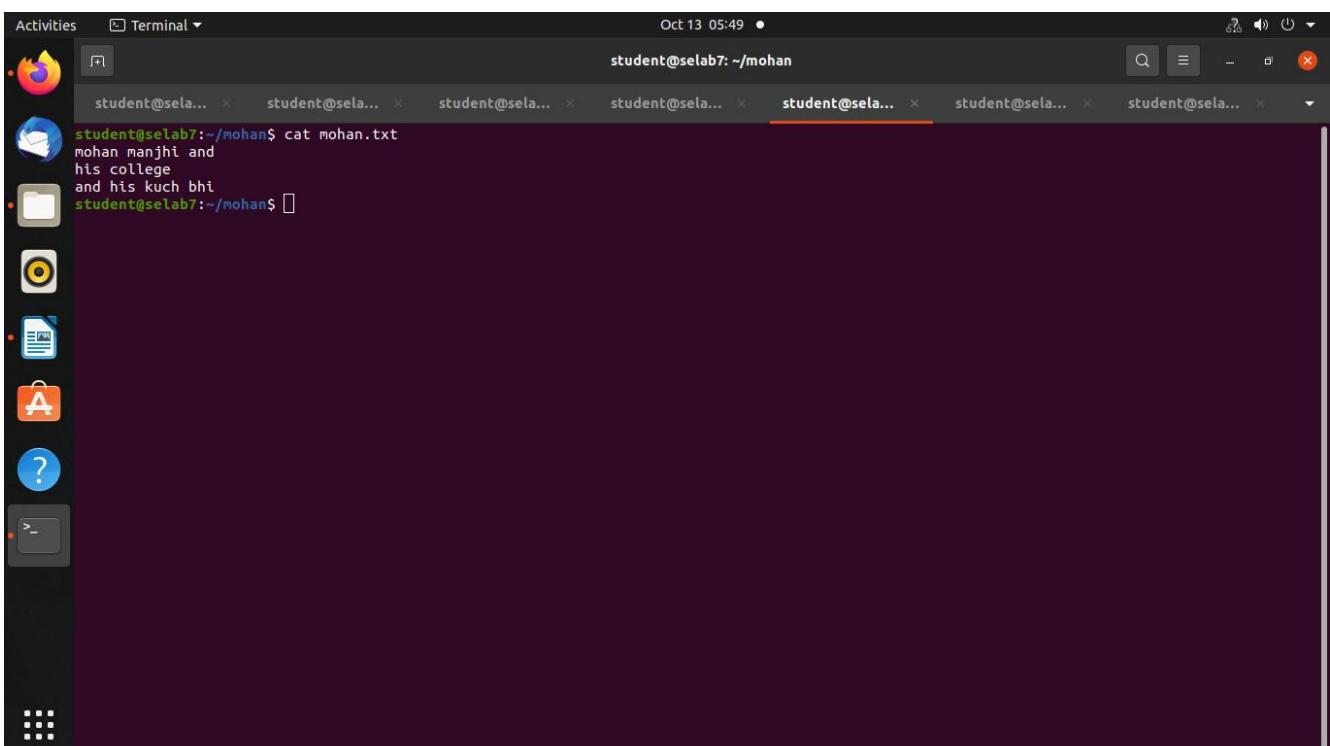
Que 4 Delete a file and remove a directory include screenshots for both actions.



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "student@selab7: ~/mohan". The terminal shows the following command history:

```
student@selab7:~/mohan$ ls
del      'Linux Assignment_mohan.docx'  mohan2.txt  mohan.txt  newlinux.txt
delfile.txt  LinuxAssignment.png        mohan3.txt  newFolder  'text file.png'
student@selab7:~/mohan$ rm delfile.txt
student@selab7:~/mohan$ rmdir del
student@selab7:~/mohan$ ls
'Linux Assignment_mohan.docx'  LinuxAssignment.png  mohan2.txt  mohan3.txt  mohan.txt  newFolder  newlinux.txt  'text file.png'
student@selab7:~/Mohan$
```

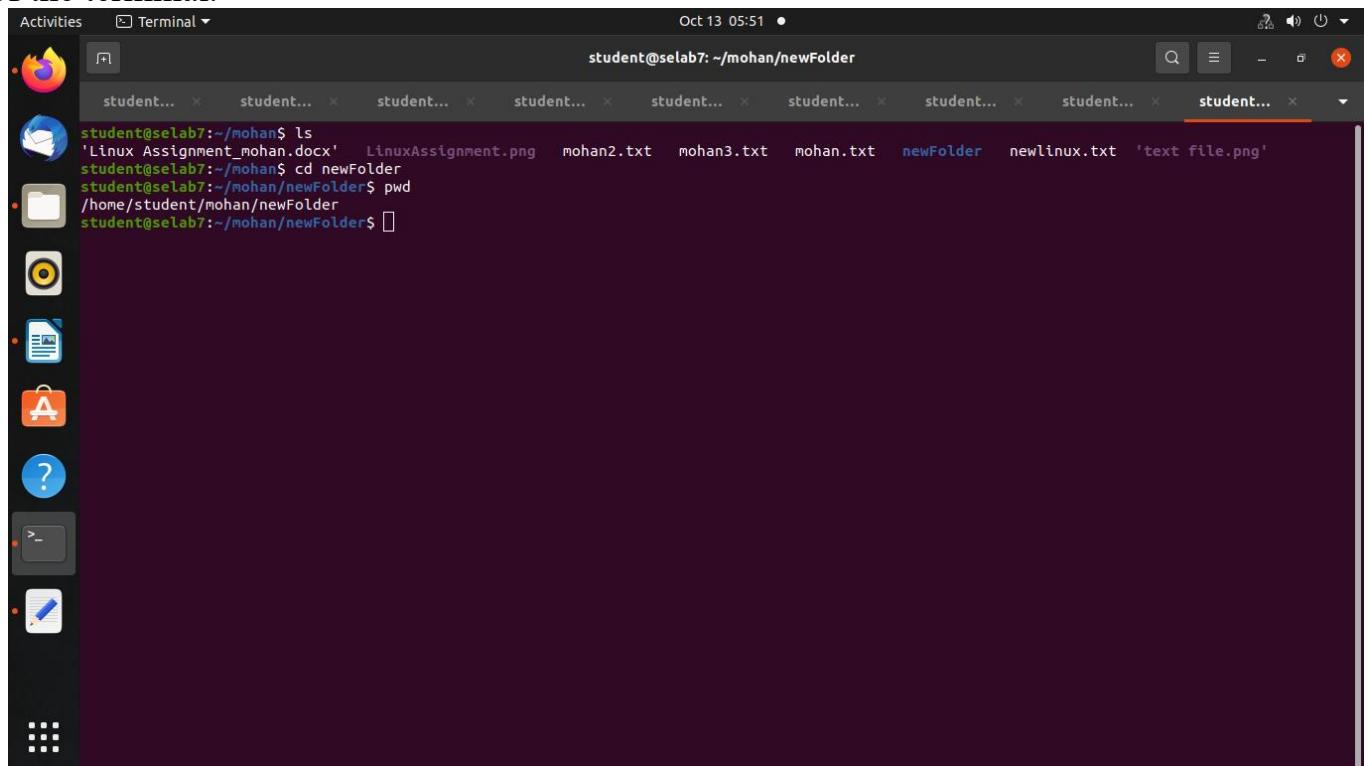
Que 5 Execute the command which display the contents of a file captures a screenshot showing the text.



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "student@selab7: ~/mohan". The terminal shows the following command history:

```
student@selab7:~/mohan$ cat mohan.txt
mohan manjhi and
his college
and his kuch bhi
student@selab7:~/mohan$
```

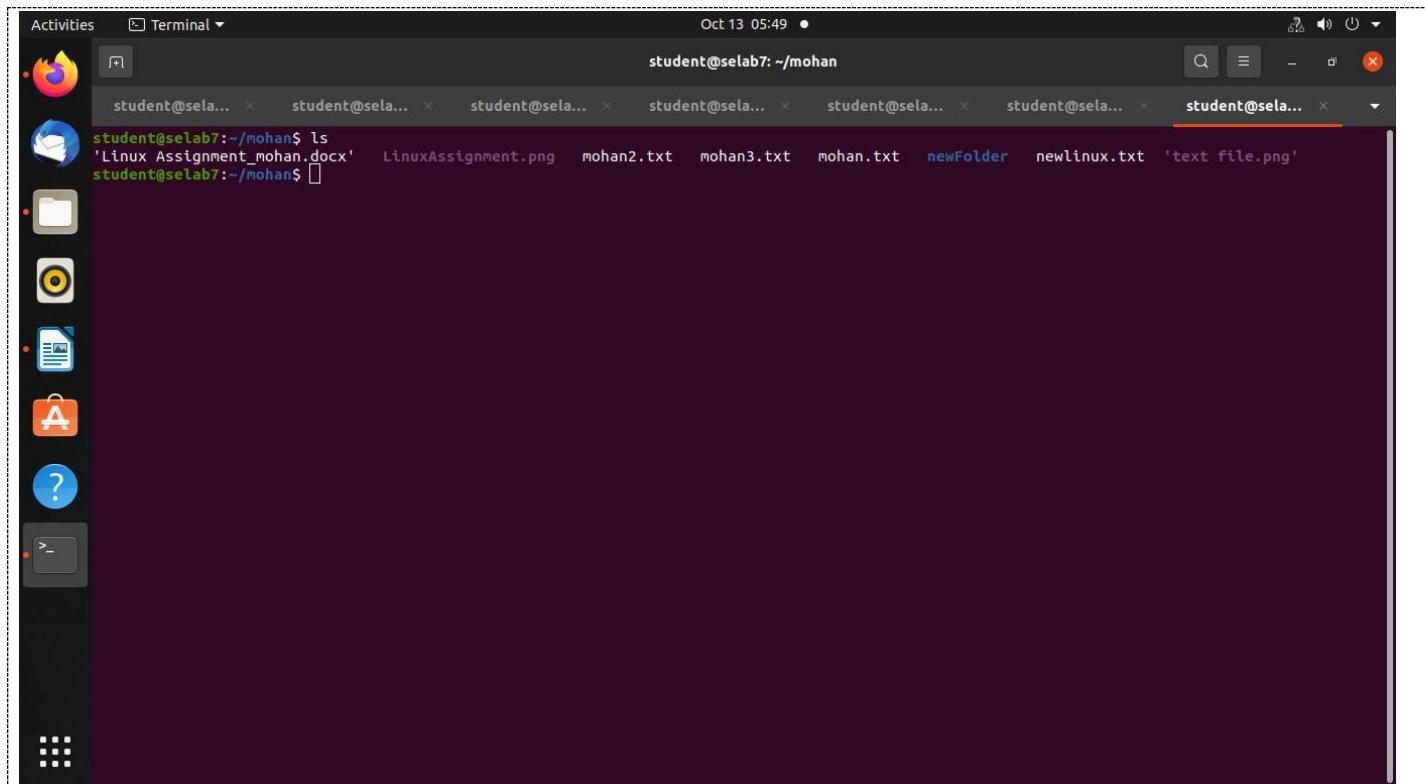
Que 6 Execute the command to navigate to a directory of your choice and take a screenshot of the terminal.



A screenshot of a Linux desktop environment. On the left is a vertical dock with icons for various applications like a browser, file manager, terminal, and system settings. The main area shows a terminal window titled "Terminal". The terminal window has a dark background and contains the following text:

```
student@selab7:~/mohan/newFolder
student@selab7:~/mohan$ ls
'Linux Assignment_mohan.docx'  LinuxAssignment.png  mohan2.txt  mohan3.txt  mohan.txt  newFolder  newlinux.txt  'text file.png'
student@selab7:~/mohan$ cd newFolder
student@selab7:~/mohan/newFolder$ pwd
/home/student/mohan/newFolder
student@selab7:~/mohan/newFolder$
```

Que 7 Execute the command in your terminal displaying the list of files and directories in your current location.



## Displaying Directories using “ ll “ commannd

A screenshot of a terminal window titled "cndc-14@cndc14-OptiPlex-3050-AIO: ~/Documents". The user runs the command "cd Documents" followed by "ls" and "ll". The output shows a list of files and directories in the "Documents" folder. The "ll" command provides detailed file information including permissions, owner, group, size, and modification date. The output is as follows:

```
cndc-14@cndc14-OptiPlex-3050-AIO: ~
cndc-14@cndc14-OptiPlex-3050-AIO: ~/Documents
cndc-14@cndc14-OptiPlex-3050-AIO: ~$ cd Documents
cndc-14@cndc14-OptiPlex-3050-AIO: ~/Documents
'aakkAND ' 'computer arc.odt' 'not gate' 'report33 (4 files merged).pdf' 'sarthak.odt'
'and gate' 'halfadder' 'ornand.logic' 'sapna' 'shreyash assingment no.1 cndc.odt'
cndc-14@cndc14-OptiPlex-3050-AIO: ~/Documents
total 2488
drwxr-xr-x  4 cndc-14 cndc-14  4096 Sep 26 15:58  /
drwxr-x--- 24 cndc-14 cndc-14  4096 Oct 17 16:39  ../
drwxrwxr-x  2 cndc-14 cndc-14  4096 Aug 28 15:34  'aakkAND '
-rw-rw-r--  1 cndc-14 cndc-14   3751 Sep  4 12:05  'and gate'
-rw-rw-r--  1 cndc-14 cndc-14 1598356 Sep 26 15:58  'computer arc.odt'
-rw-rw-r--  1 cndc-14 cndc-14   6734 Sep  4 12:44  'halfadder'
-rw-rw-r--  1 cndc-14 cndc-14    94 Sep 26 15:58  '~-lock.computer arc.odt#'
-rw-rw-r--  1 cndc-14 cndc-14  3747 Sep  4 11:53  'not gate'
-rw-rw-r--  1 cndc-14 cndc-14  4295 Aug 28 14:52  'ornand.logic'
-rw-rw-r--  1 cndc-14 cndc-14 445011 Sep 26 13:51  'report33 (4 files merged).pdf'
drwxrwxr-x  2 cndc-14 cndc-14  4096 Sep  4 12:11  'sapna/'
-rw-rw-r--  1 cndc-14 cndc-14 253277 Aug 10 13:07  'sarthak.odt'
-rw-rw-r--  1 cndc-14 cndc-14 192658 Aug 22 12:55  'shreyash assingment no.1 cndc.odt'
-rw-rw-r--  1 cndc-14 cndc-14  3747 Sep  4 11:51  'yourFile.logic'
cndc-14@cndc14-OptiPlex-3050-AIO: ~/Documents
```

**\* Movie Ticket Booking \***

**Mini Project in C**

# Mini Project Overview

**Project Title:** Movie Ticket Booking System in C

## Intruduction :

- The Movie Ticket Booking System in C is a command line application written in c.
- It allows to users to Book movie ticket and Allow to admin check ticket booking.

## Project Description:

Simple Movie Ticket Booking project is based on the concept of booking movie tickets.

There's no login system available for this project, the user can freely use its feature. This mini project contains limited features, but the essential one.

Talking about the features of this Simple project, the user can book movie tickets. For this, the user has to select a movie name, enter customer details such as name and phone number.

Then the user has to enter seat number. After this, the booking is done. The Admin can also change the ticket price and view reservations by entering the admin password.

The last feature of this project is about canceling the tickets which can be done by entering the booking id.

The system does not create an external file to store the user's data permanently. This system is in C Programming Language and different variables, strings have been used for the development of it.

The admin password is : mohan

## Features:

1. Booking tickets
2. Cancel tickets
3. Change ticket price
4. View all booking records

## Source Code

```
#include<stdio.h>
#include<stdlib.h>
#include<strings.h>
#include<string.h>

struct moviedetails{
    char name[25];
    char phone[15];
    int seat;
    int id;
};

int changeprize(int);
void reservation(int *,int,int );
int choice1(void);
void cancel(int *);
void ticket1(int choice,char name[10],int id2,int price);
void ticket2(int choice,char name[10],int id2,int price);
void ticket3(int choice,char name[10],int id2,int price);
int cmovie(void);
int movie(void);
void details(void);

struct moviedetails person[300];
int count=0;
int id2=1000;
```

```

int main()
{
    int **seat,choice,price=500,slection,i;
    seat=(int **)calloc(101,sizeof(int *));
    for (i=0;i<3;i++)
        *(seat+i)=(int *)calloc(101,sizeof(int ));
    int x;
    while(x!=5)
    {
        choice=choice1();
        switch(choice)
        {
            case 1:
                price=changeprize(price);
                break;
            case 2:
                details();
                break;
            case 3:
                slection=movie();
                reservation(seat[slection-1],price,slection);
                count++;
                break;
            case 4:
                slection=cmovie();
                cancel(seat[slection-1]);
                break;
            case 5:
                x=5;
                break;
            default:
                printf("Choice not available\n");
                break;
        }
    }
}

int changeprize(int prize)
{
    char pass[10],pak[10]="mohan";
    printf("Enter the password to change price of ticket: ");
    scanf("%s",&pass);
    if (strcmp(pass,pak)==0)
    {
        printf("Please enter new price: ");
        scanf("%d",&prize);
    }
}

```

```

        system("PAUSE");
        system("CLS");
    }
else
    printf("The entered password is wrong! ");
return prize;
}

void reservation(int *array,int price,int slection)
{
    int i,j;
    printf("\n                  SCREEN\n\n\n");
    for (i=1;i<=100;i++)
    {
        if (array[i]==0)
            printf("%d\t",i);
        else
            printf("*\t");
        if(i%10==0)
            printf("\n\n");
    }
    printf("Please enter your name: ");
    scanf(" %19[^\\n]*[^\\n]",&person[count].name);
    printf("Please enter your phone number: ");
    scanf("%u",&person[count].phone);
    printf("Which seat number you want? ");
    scanf("%d",&j);
    if (j>100| |j<1)
    {
        printf("seat1 number is unavailable in this theater\n");
        printf("Please re-enter seat number: ");
        scanf("%d",&j);
    }
    if (array[j]==1)
    {
        printf("Sorry, this ticket is already booked! Please choose another
seat.\n");
        scanf("%d",&j);
    }
}

```

```

        }

    else
        array[j]=1;
    person[count].seat=j;
    if (slection==1)
        ticket1(j,person[count].name,id2,price);
    else if (slection==2)
        ticket2(j,person[count].name,id2,price);
    else
        ticket3(j,person[count].name,id2,price);
    id2++;
}

int choice1(void)
{
    int choice;
    printf("      Simple Movie Ticket Booking System\n");
    printf("=====\\n");
    printf(" | 1- To edit price of ticket (only admin):   ||\\n");
    printf(" | 2- To view reserved tickets (only admin): ||\\n");
    printf(" | 3- To purchase ticket:                   ||\\n");
    printf(" | 4- To cancel the seat:                  ||\\n");
    printf(" | 5- Exit system:                         ||\\n");
    printf(" | =====||\\n");
    printf(" Enter your choice: ");
    scanf("%d",&choice);
    return choice;
}
void cancel(int *array)
{
    int Cseat,i,stop;
    printf("Please enter ID number of ticket: ");
    scanf("%d",&Cseat);
    for (i=0;i<300;i++)
    {
        if(Cseat==person[i].id)
        {
            stop=5;
            system("cls");
            printf("%s your seat is %d cancelled",person[i].name,person[i].seat);
            array[person[i].seat]=0;
            i=300;
        }
    }
}

```

```

    if (stop!=5)
        printf("Ticket ID number is incorrect please enter right one to cancel ticket: \n");
}

void details(void)
{
    int i;
    char pass[10],pak[10]="mohan";
    printf("Enter the password to see details: ");
    scanf("%s",&pass);
    if (strcmp(pass,pak)==0)
    {
        for (i=0;i<count;i++)
        {
            printf("seat no: %d is booked by%s booking id
is %d\n",person[i].seat,person[i].name,person[i].id);
        }
    }
    else
        printf("Entered password is wrong \n");
    system("PAUSE");
    system("CLS");
}

int movie(void)
{
    int i;
    system("cls");
    printf("\t\t\twhich movie you want to see?\n");
    printf("\t\t\t-----\n\n");
    printf("\t\t\tpress 1 for 3 Idiots (Engineering Life)\n\n");
    printf("\t\t\tpress 2 for Lakshya \n\n");
    printf("\t\t\tpress 3 for Story of Mohan: A Engineer Student\n");
    scanf("%d",&i);
    system("cls");
    return i;
}
int cmovie(void)
{
    int i;
    printf("\t\t\twhich movie ticket you want to cancel\n");
    printf("\t\t\t-----\n");
    printf("\t\t\tpress 1 for 3 Idiots (Engineering Life)\n\n");
    printf("\t\t\tpress 2 for Lakshya\n\n");
    printf("\t\t\tpress 3 for Mohan: A Engineer Student\n");
    scanf("%d",&i);
    return i;
}

```

