# Certificate

Name: Rishikesh Kumar Class: I$^{st}$ year

Roll No: 20103051 Exam No: II$^{nd}$

Institution Guru Ghasidas University, Bilaspur

*This is certified to be the bonafide work of the student in the* C programming *Laboratory during the academic year 2021/2022.*

*No. of practicals certified* _____ *out of* _____ *in the subject of* _____

.....
*Teacher In-charge*

.....
*Examiner's Signature*

.....
*Principal*

*Date:* .....

*Institution Rubber Stamp*

*(N.B: The candidate is expected to retain his/her journal till he/she passes in the subject.)*

Date _____

Expt. No. 01 Page No. _____

Objective : Program to print "Hello GGV"

C Program :

```
#include <stdio.h>
int main() {
    printf("Hello GGV");
    return 0;
}
```

Output :

Hello GGV

Teacher's Signature _____

Objective : Program to calculate area of circle.

C program :

```
#include<stdio.h>
int main() {
float r, PI = 3.14, area;
    printf("Enter the radius of circle");
    scanf("%f", &r);

    area = PI x r x r;
    printf("Area of circle is %f", area);
    return 0;
}
```

Date _____

Expt. No. 03 Page No. _____

Objective: Program to convert fahrenheit to Celcius.

C program:

```
#include <stdio.h>

int main() {
    float f, C;
    printf("Enter temperature in fairenheit: ");
    Scanf("%f", &f);

    C = ((f-32)*5)/9;
    printf("Temperature in Celcius: %f", C);

    return 0;
}
```

Teacher's Signature _____

Objective: Program to check if the entered character is 'A' or not:

C program:

```
#include <stdio.h>

int main() {
    char c;
    printf("Enter the character: ");
    scanf("%c", &c);

    if (c == 'A') printf("Yes");
    else printf("No");

    return 0;
}
```

Expt. No. 5 Page No. Date

Objective: Check whether the number is even or odd.

C program:

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number: ");
    scanf("%d",&n);

    //General method
    if (n%2==0) printf("Even");
    else printf("Odd");

    // Bit manipulation
    if (n&1) printf("Odd");
    else printf("Even");

    return 0;
}
```

Note :- Bit manipulation programs are generally faster than normal programs. That is why it is advised to use bit manipulation as often as you can in your programs.

Teacher's Signature

Expt. No. 6 Page No.

Objective : print "Hello GGV" 10 times

C program :

```
#include <stdio.h>

int main() {
    int n=10;

    // for loop implementation
    for(int i=0; i<n; i++) printf("Hello GGV\n");

    // while loop implementation
    int i=0;
    while(i<n) printf("Hello GGV\n");

    return 0;
}
```

Note :- '\n' character is called endline character

Teacher's Signature

Expt. No. 7 Page No.

Objective: Display the last two digits of a number

C program:

```
#include <stdio.h>

int main () {
    int n;
    printf (" Enter the number: ");
    Scanf ("%d", &n);

    if (n < 10) {
        printf ("Invalid input (less than 10)");
        return 0;
    }

    printf ("last two digits are %d", (n%100));

    return 0;
}
```

Note:- If we use 'return 0' in between the program the program will not execute any further and will exit there.
This is used to ~~check~~ make program more readable by reducing the use of 'if-else' statements.

Teacher's Signature

Date

Expt. No. 8

Page No.

Objective: Print numbers from 1 to n.

C program:

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number: ");
    scanf("%d", &n);

    for(int i=1; i<=n; i++) printf("%d\n", i);

    return 0;
}
```

Note:- It is a good practise to use "for loops" when number of iteration is known.
And while loop, when you have to run a program until a certain condition.
Although both the loops can be used interchangeably.

Teacher's Signature

Date _____

Expt. No. 9 Page No. _____

Objective: Print the table of n.

C program:

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number: ");
    scanf("%d", &n);

    for (int i=0; i<=n; i++)
        printf("%d x %d = %d\n", n, i, (n*i));

    return 0;
}
```

Teacher's Signature _____

Expt. No. 10 Page No.

Objective : Find the factorial of n.

Theory : Factorial of n is $1 \times 2 \times 3 \times \ldots \times n$.

C program:

```
#include <stdio.h>

int main(){
    int n, fact = 1;
    printf(" Enter the number: ");
    scanf("%d", &n);

    for (int i=1; i<=n; i++)
        fact *= i;

    printf(" Factorial of %d is %d", n, fact);

    return 0;
}
```

Note: factorial program will give correct output upto 16 as input, when int is used as container. Because factorial becomes very big for even small inputs.
We can use 'long long int' for bigger inputs.

Teacher's Signature

Expt. No. 11 Page No.

Objective : print the sum of all natural numbers till n.

Theory: Sum of all natural numbers till n
$= \frac{n \times (n+1)}{2}$.

C program:

```
#include <stdio.h>

int main() {
    int n, sum=0;
    printf("Enter the number: ");
    scanf("%d", &n);

    // using for loop
    for(int i=1; i<=n; i++) sum += i;

    // Mathematical approach
    sum = (n * (n+1)) / 2;
    Printf("Sum till %d is %d", n, sum);
    return 0;
}
```

Note: In programming, there is a notion of time complexity.
In this question, if we use 'for loop' the time complexity is O(n), while in mathematical approach time complexity is O(1), which better.

Teacher's Signature

Expt. No. 12 Page No.

Objective: find sum of series $1^2+2^2+3^2+\cdots+N^2$.

Theory: Using sequence and series equation,

$$1^2+2^2+3^2+\cdots+N^2 = N\times(N+1)\times(2N+1)/6;$$

C program:

```
#include <stdio.h>
int main(){
    int n, sum=0;
    printf("Enter the number: ");
    scanf("%d", &n);

    // Using loop
    for(int i=1; i<=n; i++) sum += i*i;

    // Mathematical approach
    sum = (n*(n+1)*(2*n+1))/6;

    printf("Sum of series till %d is %d", n, sum);

    return 0;
}
```

Note: Time complexity
Loop implementation - O(n)
Mathematical - O(1)

Teacher's Signature

Date

Expt. No. 13

Page No.

Objective: Print the sum of series $1^3+2^3+3^3+\cdots+N^3$.

Theory: In sequence and series,

$$1^3+2^3+3^3+\cdots+N^3 = \left(\frac{N\times(N+1)}{2}\right)^2$$

C program:

```
#include <stdio.h>
int main(){
    int n, sum=0;
    printf("Enter the number: ");
    scanf("%d", &n);

    // using loop
    for(int i=1; i<=n; i++) sum += i*i*i;

    // Mathematically
     sum = (N*(N+1))/2;
     sum *= sum;

    printf("Sum of the series till %d is %d", n, sum);

    return 0;
}
```

Teacher's Signature

Date _____

Expt. No. 14 Page No. _____

Objective: Calculate $a^b$ using loops.

Theory: $a^b = \begin{cases} a \times a^{b-1}, & \text{if } b \text{ is odd} \\ (a^2)^{b/2}, & \text{if } b \text{ is even} \end{cases}$

C program:

```
#include<stdio.h>
int main() {
    int a, b, pow=1;
    printf("Enter a and b: ");
    scanf("%d %d", &a, &b);

    //loop implementation
    for(int i=1; i<=b; i++) pow *= a;

    //fast power method
    while(b) {
        if(b&1) {
            pow *= a;
            b--;
        } else {
            pow *= pow;
            b /= 2;
        }
    }
    printf("%d to power %d is %d", a, b, pow);

    return 0;
}
```

Teacher's Signature _____

Expt. No. 15 Page No.

Objective : Check whether the person is eligible for voter ID card.

Law : If a person is citizen of india and his/her age is greater than or equal to 18 yrs. Then he/she is eligible for voter ID card.

C program:

```
#include<stdio.h>
int main(){
    int age; char nationality[100];
    String nationality;
    printf("Enter your age: ");
    scanf("%d", &age);
    printf("Enter your nationality: ");
    scanf("%s", &nationality);

    if(age >= 18 && nationality.strcmp("Indian")==0)
        cout << "you are eligible\n";
    else
        cout << "you are not eligible\n";
    return 0;
}
```

Teacher's Signature

Expt. No. 16 Page No.

Objective : Calculate the average of 5 numbers.

C program:

```
#include <stdio.h>
int main()
    float a,b,c,d,e, avg;
    printf(" Enter five numbers: ");
    scanf("%f %f %f %f %f",&a,&b,&c,&d,&e);

    avg = (a+b+c+d+e)/5.
    printf("The average is: %d", avg);
    return 0;
}
```

Teacher's Signature

Date _____

Expt. No. 17 Page No. _____

Objective: Swap two numbers using 3 variables.

C program:

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

    temp = a;
    a    = b;
    b    = temp;

    printf("Values after swapping\n a: %d\n b: %d", a, b);
    return 0;
}
```

Teacher's Signature _____

Date _____

Expt. No. 18 Page No. _____

Objective: Swap two numbers 2 variables only.

C program:

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

      a = a^b;
      b = a^b;
      a = a^b;

    printf("values after swapping: \n");
    printf("a = %d\nb = %d", a, b);

    return 0;
}
```

Teacher's Signature _____

Expt. No. 19 Page No.

Objective: Print character using ASCII value.

Theory: ASCII stands for:
"American Standard Code for Information Interchange"

'a'-'z' = 97-122
'A'-'Z' = 65-90
'0'-'9' = 48-57

There are 256 ASCII values.

C program:

```
#include<stdio.h>
int main() {
    int a;
    char c;
    printf("Enter the ASCII code: ");
    scanf("%d", &a);

    c = a;

    printf("The character is %c", c);

    return 0;

}
```

Teacher's Signature

Date

Expt. No. 20

Page No.

Objective: Print ASCII using character.

C program:

```
#include<stdio.h>
int main(){

    char c;
    printf("Enter the character: ");
    scanf("%c", &c);

    printf("The ASCII code is %d", c);

    return 0;
}
```

Teacher's Signature

Expt. No. 21 Page No.

Objective: Check for vowel or consonant.

C program:

```
#include <stdio.h>
int main() {
    char c;
    printf("Enter an alphabet: ");
    scanf("%c", &c);

    if (c=='a' || c=='A') || c=='e' || c=='E')
     || (c=='i' || c=='I') || (c=='o' || c=='O')
     || (c=='u' || c=='U')
        printf("Character is vowel");
    else printf("Character is consonant");

    return 0;

}
```

Teacher's Signature

Expt. No. 22 Page No.

Objective: Check whether triangle is equilateral, Isosceles or scalene.

Theory: Equilateral triangle – all sides equal.
Isosceles triangle – two sides equal one unequal.
scalene triangle – all sides unequal.

C program:

```
#include <stdio.h>
int main() {
    float s1, s2, s3;
    printf("Enter the sides of triangle: ");
    scanf("%f %f %f", &s1, &s2, &s3);

    if(s1==s2 && s2==s3) printf("Equilateral");
    else if(s1==s2 || s2==s3 || s3==s1)
        printf("Isosceles");
    else printf("scalene");

    return 0;
}
```

Teacher's Signature

Expt. No. 23 Page No.

Objective: check for leap year.

Theory: If a year is divisible by 4 then the year is leap year.

C program:

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

    if(year%4==0) printf("leap year");
    else printf("not a leap year");

    return 0;
}
```

Teacher's Signature

Expt. No. 24 Page No.

Objective: Swap cases of the string.

C program:

```
#include <stdio.h>
int main() {
    ~~char str[50];~~
    char c;
    printf("Enter the character: ");
    scanf("%c", &c);

    if (c >= 'a' && c <= 'z') {
        c -= 32;
        printf("character is lowercase\n");
        printf("It's uppercase is: %c", c);
    }
    else if (c >= 'A' && c <= 'Z') {
        c += 32
        printf("character is uppercase\n");
        printf("It's lowercase is: %c", c);
    }

    return 0;
}
```

Teacher's Signature

Objective: Swap cases of the string.

C program:

```
#include <stdio.h>
int main() {
    char str[50];
    char c;
    printf("Enter the character: ");
    scanf("%c", &c);

    if (c >= 'a' && c <= 'z') {
        c -= 32;
        printf("character is lowercase");
        printf("It's uppercase is %c", c);
    }
    else if (c >= 'A' && c <= 'Z') {
        c += 32;
        printf("character is uppercase\n");
        printf("It's lowercase is %c", c);
    }

    return 0;

}
```

Teacher's Signature _____

Expt. No. 25 Page No.

Objective: Print numbers using "continue" and "break" statements.

C program:

```
#include <stdio.h>
int main(){

    //Printing numbers multiple by 3 or 5 upto n
    int n, i=0;
    printf("Enter the number: ");
    scanf("%d", &n);

    while(1){
        if(i%3==0 || i%5==0) printf("%d\n", i);
        else if(i>n) break;
        else continue;

    }

    return 0;

}
```

Teacher's Signature

Date

Expt. No. 26 Page No.

Objective : Calculate HCF of two numbers.

C program:

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

    int i=2;
    for(i=2; i<=a; i++) if(a%i==0 && b%i==0) break;

    printf("The HCF of two numbers is %d", i);

    // Another approach

    while(a>0){
        int temp = a;
        a = a%b;
        b = temp;
    }
    printf("The HCF is: %d", b);

    return 0;
}
```

Teacher's Signature

Date

Expt No. 27 Page No.

Objective: Calculate LCM of two numbers:

Theory: for two numbers a and b.
LCM(a, b) × HCF(a, b) = a × b

C program:

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

    int temp1, temp2, temp;
    temp1 = a, temp2 = b;
    while (temp1 > 0) {
        temp = a; temp1;
        a = a % b;
        b = a;
    }

    lcm = (temp1 * temp2) / b;

    printf("LCM is: %d", lcm);

    return 0;
}
```

Teacher's Signature

Expt. No. 28 Page No.

Objective : Print sum and reverse order.

C program :

```
#include <stdio.h>
int main() {
    int n, sum=0, rev=0;
    printf("Enter the number: ");
    scanf("%d", &n);

    while(n>0){
        rev = rev*10 + (n%10);
        sum +=(n%10);
        n /=10;
    }

    printf("The sum of digits is: %d" sum);
    printf("The reverse is: %d" rev);

    return 0;
}
```

Teacher's Signature

Expt No. 29 Page No.

Objective: check for palindrom.

Theory: A number is called palindrom if its reverse is equal to the original number.
for example
1234 is not a palindrom. [since 1234 ≠ 4321]
121 is a palindrom. [since 121 = 121]

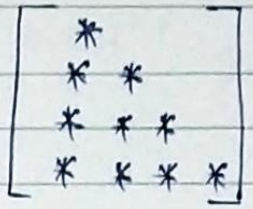
C program:

```
#include <stdio.h>
int main() {
    int n, rev = 0;
    printf("Enter the number: ");
    scanf("%d", &n);

    int temp = n;
    while(temp >0) {
        rev = ~~ten~~ rev*10 + (temp%10);
        temp /= 10;
    }

    if(n == rev) printf("The number is palindrom");
    else printf("The number is not a palindrom");

    return 0;
}
```

Teacher's Signature

Date _____

Expt. No. 30 Page No. _____

Objective: Print pattern  and so on...

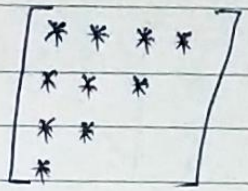
C program:

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

    for(int i=0; i<n; i++){
        for(int j=0; j<=i; j++) printf("* ");
        printf("\n");
    }

    return 0;

}
```

Teacher's Signature _____

Date _____

Expt. No. 31 Page No. _____

Objective : Print pattern  and so on...

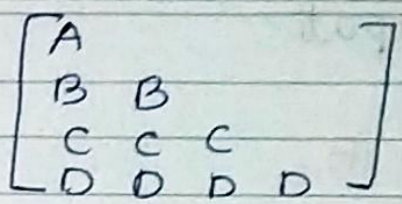
C program:

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

    for(int i=0; i<n; i++){
        for(int j=0; j<n-i; j++) printf("* ");
        print("\n");
    }

    return 0;

}
```

Teacher's Signature _____

Expt. No. 32 Page No.

Objective: Print the pattern




C program:

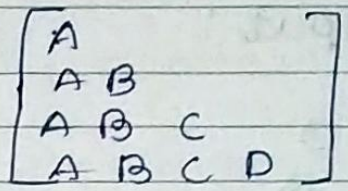
```
#include<stdio.h>
int main(){
    int n;
    printf("Enter the number of row: ");
    scanf("%d",&n);

    for(int i=0; i<n ; i++){
        for(int j=0; j<=i; j++) printf("%c ",'A'+i);
        printf("\n");

    }

    return 0;

}
```

Teacher's Signature

Date _____

Expt. No. 33 Page No. _____

Objective: print the pattern




C program:

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

    for(int i=0; i<n; i++){
        for(int j=0; j<=i; j++) printf("%c", 'A'+j);
        printf("\n");
    }

    return 0;

}
```

Teacher's Signature _____

Date

Expt. No. 34 Page No.

Objective: find an armstrong number

Theory: Armstrong number is a number whose sum of cubes of digits is equal to the original number.

For example:

$153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

C program:

```
#include<stdio.h>
int main() {
    int n, sum=0;
    printf("Enter the number: ");
    scanf("%d", &n);

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

    if(sum==n) printf("Armstrong number");
    else printf("Not an Armstrong number");

    return 0;

}
```

Teacher's Signature

Date _____

Expt. No. 35 Page No. _____

Objective: Count the number of digits.

C program:

```
#Include <stdio.h>
int main() {
    int n, count = 0;
    printf(" Enter the number: ");
    &canf("%d", &n);

    while(n>0) {
        count++;
        n/=10;
    }

    // Alternate method
    count = log10(n) + 1;

    printf("The number of digits is: %d", count);

    return 0;

}
```

Teacher's Signature _____

Expt. No. 36 Page No.

Objective: Find even or odd using bit manipulation.

Theory: In odd number, the least significant is a set bit, while in even number it's not true.
for example.
5 = 101
4 = 100

C program.

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

    if(n&1) printf("Odd\n");
    else printf("Even\n");

    return 0;

}
```

Teacher's Signature

Objective: Minimum bits required to store the number.

C program:

```
#include <stdio.h>
int main () {
    int n, count = 0;
    printf (" Enter the number: ");
    scanf (" %d", &n);

    while (n > 0) {
        count++;
        n >>= 1;
    }

    printf (" Bits required = %d", count );

    return 0;

}
```

Objective : Count number of ones (1's).

C program:

```
#include <stdio.h>
int main() {
    int n, count = 0;
    printf("Enter the number: ");
    scanf("%d", &n);

    for (int i=0; i<32; i++) {
        count += (n & (1<<32)) != 0;
    }

    // Alternate method
    while (n) {
        n = n & (n-1);
        count++;
    }
    printf("The number of 1's is %d", count);
    return 0;

}
```

Teacher's Signature

Expt. No. 39 Page No.

Objective: Check whether the position is set or not.

Theory: In binary representation of numbers, a position is said to be set if it is 1.
For example:
10 = 1010, underlined are set bit

C Implementation:

```
#include <stdio.h>
int main() {
    int n, pos;
    printf("Enter the number: ");
    scanf("%d", &n);
    printf("Enter the position: ");
    scanf("%d", &pos);

    if (n & (1<<pos)) printf("Position is set bit");
    else printf("position is not set bit");

    return 0;

}
```

Teacher's Signature

Date

Expt. No. 40

Page No.

objective : Initialise and print the array.

Theory: Array is a data structure. It is continuous and accessing is very fast O(1).

C program:

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

    int arr[n];
    printf("Enter the array elements");
    for(int i=0; i<n; i++){
        scanf("%d", &arr[i]);
    }

    for(int i=0; i<n; i++){
        printf("%d ", arr[i]);
    }

    return 0;
}
```

Teacher's Signature

Date

Expt. No. 41 Page No.

Objective: Check whether letter is vowel or consonant in an array.

C program:

```
#include <stdio.h>
int main() {
    int n; arr[n];
    printf("Enter number of characters
```

```
int main() {
    int n;
    char arr[n];
    printf("Enter the number of characters: ");
    scanf("%d", &n)
    printf("Enter the characters: ");
    Char arr[n];
    for(int i=0; i<n; i++) scanf("%c", &arr[i]);

    for(int i=0; i<n; i++) {
        if(arr[i]=='a'||arr[i]=='e'||arr[i]=='i'||
          arr[i]=='o'||arr[i]=='u'||arr[i]='A'||
          arr[i]=='E'||arr[i]=='I'||arr[i]='O'||
          arr[i]=='U') printf("Vowel ");
        else consonant printf("consonant ");
    }

    return 0;
}
```

Teacher's Signature

Expt. No. 42 Date

Page No.

Objective: Calculate average of five numbers in an array

C program:

```
#include <stdio.h>
int main() {
    int n=5;
    float arr[n];
    printf("Enter the elements: ");
    for(int i=0; i<n; i++) scanf("%f", &arr[i]);

    float sum=0;
    for(int i=0; i<n; i++) sum += arr[i];

    float avg = sum/5;

    printf("Average is %f", avg);

    return 0;

}
```

Teacher's Signature

Date _____

Expt. No. 43 Page No. _____

objective: check odd or even an array.

C program:

```
#include <stdio.h>
int main(){
    int n; arr[n];
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter the elements: ");
    for(int i=0; i<n; i++) scanf("%d", &arr[i]);

    for(int i=0; i<n; i++){
        if(arr[i]&1) printf("Odd ");
        else printf("Event ");
    }

    return 0;
}
```

Teacher's Signature _____

Expt. No. 44 Page No.

Objective: Reverse the array.

C program:

```
#Include <stdio.h>
int main () {
    int n; 
    printf (" Enter the number of elements: ");
    scanf ("%d", &n);
    printf (" Enter the elements : ");
  int arr[n];
    for(int i=0; i<n; i++) scanf ("%d", &arr[i]);

    int i=0, j = n-1;
    while (i<j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
        i++;
        j--;
    }

    printf ("Array after reversing :\n")

    for(int i=0; i<n; i++) printf ("%d ", arr[i]);

    return 0;

}
```

Teacher's Signature

Expt. No. 45 Page No.

Objective: Calculate size of array:

C program:

```
#include <stdio.h>
int main() {
    int n, arr[n];
    int arr[] = {1, 2, 3, 4, 5};
    printf("length of array is %d", sizeof(arr)/sizeof(arr[0]));
    return 0;
}
```

Teacher's Signature

Date _____

Expt. No. 46 Page No. _____

Objective : print 2D array.

C program:

```
#include <stdio.h>
int main() {
    int n, m ;
    printf("Enter rows and columns: ");
    scanf("%d %d", &n, &m);
    printf("Enter the elements:");
    int arr[n][m];
    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++)
            scanf("%d", &arr[i][j]);

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

    return 0;

}
```

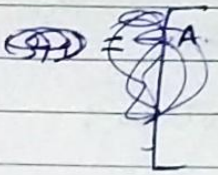
Teacher's Signature _____

Date _____

Expt. No. 47 Page No. _____

Objective: Addition of 2D matrices.

Theory: Let A(3,4) and B(3,4) be two matrices then, C(3,4) = A(3,4) + B(3,4)




C(i,j) = A(i,j) + B(i,j)

C program:

```
#include<stdio.h>
int main() {
    int n, m;
    printf("Enter row and column: ");
    scanf("%d %d", &n, &m);
    int arr[n][m], brr[n][m];
    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++)
            scanf("%d", &arr[i][j]);

    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++)
            scanf("%d", &brr[i][j]);

    int ans[n][m];

    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++)
            ans[i][j] = arr[i][j] + brr[i][j]
```

Teacher's Signature _____

Date _____

Expt. No. _____ Page No. _____

```
printf("New array after addition is :\n");

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

return 0;

}
```

Teacher's Signature _____

Expt. No. 48 Page No.

Objective: Print transpose of 2D matrix

Theory:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \qquad A^T = \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}$$

C program:

```
#include <stdio.h>
int main(){
    int n, m;
    printf("Enter row and column: ");
    scanf("%d %d", &n, &m);

    int A[n][m], AT[m][n];
    printf("Enter the elements:\n");
    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++) scanf("%d", &A[i][j]);

    for(int j=0; j<n; j++)
        for(int j=0; j<m; j++) AT[j][i] = A[i][j];

    printf("Elements in transpose array:\n");
    for(int j=0; i<n; i++){
        for(int j=0; j<m; j++) printf("%d ", AT[i][j]);
        printf("\n");
    }

    return 0;
}
```

Teacher's Signature

Expt. No. 49 Date

Page No.

Objective: Factorial of a number using do-while

Theory: Do-while loops are slightly different than for loop and while loop in the context that do-while loop executes at least once. It runs first and then checks for the condition unlike other two

C program:

```
#include <stdio.h>
int main() {
    int n, fact = 1;
    printf("Enter the number: ");
    scanf("%d", &n);

    int i = 1;
    do {
        fact *= i; i++;
    } while (i<=n)

    printf("The factorial of %d is %d", n, fact);

}
```

Terminal

```
Enter the number: 5
The factorial of 5 is 120.
```

Teacher's Signature

Expt. No. 50 Page No.

Objective: Code encryption

Theory: Encryption is a method of data hiding. In this code we will print the next character to the corresponding character in array.
Eg. - "sumit" → "tvnju";

C program:

```
#include <stdio.h>
int main() {
        char name[50];
        printf("Enter the name: ");
        scanf("%[^\n]%c", &name);

        int i=0;
        while (name[i] != '\0') {
                name[i]++;
                i++;
        }

        printf("The name after encryption is %s", &name);
}
```

Terminal

Enter the name: Prince Kumar
The name after encryption is Qsjodg!Lvnbs

Teacher's Signature

Expt. No. 51 Page No.

Objective: Enter and display the string.

Theory: String is a continuous set of character joined together. There is no such data type as string in C but we can use the notion of string with character array.

C program:

```
#include<stdio.h>
int main(){
    char str[30];
    printf("Enter the string:");
    scanf("%[^\n]*c", &str);

    printf("The entered string is %s", str);

    return 0;
}
```

Terminal

Enter the string: Prince Kumar
The entered string is Prince Kumar.

Teacher's Signature

Expt. No. 52. Page No.

Objective: Calculate length of string without space.

Theory: In C character arrays (string) ends with a null character '\0', we can use this null character to find the length of our string.

C program:

```
#include <stdio.h>
int main() {
    char name[30];
    printf("Enter the name: ");
    scanf("%[^\n]*c", &name);

    int i=0, len=0;
    while(name[i] != '\0') len += (name[i] != ' ');

    while(name[i] != '\0') len += (name[i++] != ' ');

    printf("The length of the string is %d", len);
}
```

Terminal

```
Enter the name: Prince Kumar
The length of the string is 11.
```

Teacher's Signature

Expt. No. 53 Page No.

Objective : Convert the lowercase string to upper case.

Theory : In ASCII code value the difference between corresponding upper and lower case characters is 32.
So, to make for lowercase to uppercase we subtract 32 from the ASCII value of given character.

C program:

```
#include <stdio.h>
int main(){
    char name[30];
    printf("Enter the string: ");
    scanf("%[^\n]*c", &name);

    int i = -1;
    while(name[++i] != '\0')
        if(name[i] >= 'a' && name[i] <= 'z')
            name[i] -= 32;

    printf("The name in uppercase is %s", name);

    return 0;
}
```

Terminal

```
Enter the string: Prince Kumar
The name in uppercase is PRINCE KUMAR.
```

Teacher's Signature

Expt. No. 54 Date

Page No.

Objective: Convert even places of a string to uppercase.

C program:

```
#include<stdio.h>
int main(){
    char name[30];
    printf("Enter the name: ");
    scanf("%[^\n]%c", &name);

    int i = -1;
    while(name[++i] != '\0')
        if(i%2==0 && name[i]>='a' && name[i]<='z')
            name[i] -= 32;

    printf("The name in uppercase alternate is %s", name);

    return 0;
}
```

Terminal

```
Enter the name: Prince kumar
The name in uppercase alternate is PrInCe kUmAr
```

Teacher's Signature

Expt. No. 55 Page No.

Objective : Find a number in array.

Theory: Linear search is a method for finding an element present in an array by checking one by one each element.
Ω(1), O(n)

C program:

```
#include <stdio.h>
int main(){
    int arr[] = {1,9,8,7,2,3};
    int k;
    printf("Enter the number to find: ");
    scanf("%d",&k);

    int found = -1;
    for(int i=0; i<n; i++)
        if(k==arr[i]){
            found = i;
            break;
        }

    if(found != -1) printf("The number is at %d", found);
    else printf("Not present");

    return 0; }
```

Terminal

```
Enter the number to find: 7
The number is at 3.
```

Teacher's Signature

Expt. No. 66 Page No.

Objective: Change the third bit of a number.

Theory: To invert a particular bit we use XOR(⊕) operator

$a \oplus b = a'b + ab'$

C program:

```
#Include <stdio.h>
Int main () {
    int n;
    printf ("Enter the number: ");
    scanf ("%d", &n);

    Int mask = 1<<3;
     n ^= mask

    printf (" The number after changing the bit is %d", n);
    return 0;
}
```

Terminal

Enter the number: 7
The number after changing the bit is 15.

Teacher's Signature

Expt. No. 57 Page No.

Objective: Print A to Z except D, J, M, T and Y.

Theory: Break — breaks the loop and comes out of it.
Continues — skips the iteration and goes to next one iteration.

C program:

```
#include <stdio.h>
int main() {

    for(char c='A'; c<='Z'; c++){
        if(c=='D'||c=='J'||c=='M'||c=='T'||
            c=='Y') continue
        printf("%c ",c);

    }
    return 0;
}
```

Terminal

```
A B C E F G H I K L N O P Q R S U V W X Z
```

Teacher's Signature

Expt. No. 58 Page No.

Objective: Solve the following equation:

$$T = a! \times a \wedge b$$

Theory: $a!$ means $a \times a-1 \times a-2 \ldots 1$

$a \wedge b$ means $a \times a \times a \ldots b$ times

C Program:

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

  int fact = 1;
  for(int i=1; i<=a; i++) fact *= i;

  int pow = 1;
  while(b) {
      if(b&1) {
          pow *= a;
          b--;
      } else {
          a *= a; b/=2; }
  }
  printf("The solution of the equation is %d", fact*pow);
  return 0; }
```

Terminal

```
Enter two number: 1 2
The solution of the equation is 1
```

Teacher's Signature

Date _____

Expt. No. 39 Page No. _____

Objective: Display the pattern

A
A 1
A 1 C
A 1 C 3
A 1 C 3 E

C program:

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

    for(int i=0; i<n; i++) {
        for(int j=0; j<=i; j++) {
            if(j%2) printf("%d", j);
            else printf("%c", j+'A');
        }
        printf("\n");
    }
    return 0;
}
```

Terminal

```
Enter the number of rows: 5
A
A1
A1C
A1C3
A1C3E
```

Teacher's Signature _____

Expt. No. 60 Page No.

Objective: Sort an array using bubble sort.

Theory: Bubble sort in an sorting algorithm which compares the two consecutive elements and swap them according to the condition.

C program:

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

    int arr[n];
    printf("Enter the elements: ");
    for(int i=0; i<n; i++) scanf("%d", &arr[i]);

    for (int i=0; i<n; i++) {
        for(int j=0; j<n-i-1; j++){
            if (arr[j] > arr[j+1]) {
                arr[j] = arr[j] ^ arr[j+1];
                arr[j+1] = arr[j] ^ arr[j+1];
                arr[j] = arr[j] ^ arr[j+1];
            }
        }
    }
    printf("The sorted array is: ");
    for(int i=0; i<n; i++) printf("%d ", arr[i]);
    return 0;
}
```

Teacher's Signature

Date

Expt. No. 61

Page No.

Objective: Display "Hello GGV" using function.

Theory: Functions (also called 'procedures' and 'methods') are a set of instructions bundled together to achieve a specific outcome.

C program:

```
#include <stdio.h>

void display() { printf("Hello GGV\n"); }

int main() {
    display();
    return 0;
}
```

Terminal

```
Hello GGV
```

Teacher's Signature

Expt. No. 062 Page No.

Objective: Display the factorial of a number using functions.

C program

```
#include <stdio.h>
int factorial(int n) {
    int fact = 1;
    for(int i=0; i<=n; i++) fact *= i;
    return fact;
}

int main() {
    int n;
    printf("Enter the number: ");
    scanf("%d", &n);

    printf("The factorial of %d is %d", n, factorial(n));

    return 0;
}
```

Terminal

```
Enter the number: 5
The factorial of 5 is 120
```

Teacher's Signature

Objective: Sum two numbers using functions.

C program:

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

    printf("The sum of %d and %d is %d", a, b, add(a,b));
    return 0;
}
int add(int a, int b) { return a+b; }
```

Terminal

```
Enter two numbers: 2 3
The sum of 2 and 3 is 5.
```

Teacher's Signature

Expt. No. 64 Page No.

Objective : Display upto Z.

C program:

```
#include <stdio.h>
void display(char);

int main() {
    char c;
    printf("Enter the character: ");
    scanf("%c", &c);
    display(c);

    return 0;
}
void display(char c) {
    while (c <= 'Z') printf("%c ", c++);
}
```

Terminal

```
Enter the character: S
S T U V W X Y Z
```

Teacher's Signature

Expt. No. 65 Page No.

Objective: Area of circle using function

Theory: Area of circle = $\pi r^2$.
In C header file math.h contains M_PI.

C program:

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

double area(double);

int main(){
    double r;
    printf("Enter the radius: ");
    scanf("%lf", &r);

    printf("The area of circle is %lf", area(r));
    return 0;
}
double area(double r){return M_PI*r*r;}
```

Terminal

```
Enter the radius: 5
The area of circle is 78.539816
```

Teacher's Signature

Expt. No. 66 Page No.

Objective: Find nature and roots of quadratic equation.

Theory: $ax^2+bx+c$ is a quadratic equation

$D = b^2 - 4ac$

If $D < 0$, no real roots

If $D = 0$, both roots real and equal

If $D > 0$, real and equal.

$$x = \frac{-b \pm \sqrt{D}}{2a}$$

C program

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

int main() {
    int a, b, c;
    printf("Enter the constants of equation. ");
    scanf("%d, %d %d", &a, &b, &c);

    int D = b*b - 4*a*c;
    if (D < 0) {
        printf("The roots are unreal\n");
        printf("The roots are (-%d + %di)/(2*%d) and
            (-%d - %di)/(2*%d)", b, D, a, b, D, a);
    } else if (D == 0) {
        printf("The roots are real and equal\n");
        printf("The root will be %d", -b/2*a);
    } else {
        printf("The roots are real and unequal\n");
        printf("The roots are %d and %d", (-b + sqrt(D))/2*a),
            (-b - sqrt(D))/2*a);
    }
    return 0;
}
```

Teacher's Signature

Expt. No. 67 Page No.

Objective: Find the sum of n natural numbers using pointers and function.

Theory: The pointer * in language C is a variable which stores the address of another variable.

C program:

```
#include <stdio.h>
int sum(int n);

int main() {
    int n;
    printf("Enter the number: ");
    scanf("%d", &n);

    printf("The sum of %d natural numbers is %d", n, sum(n));
    return 0;
}
int sum(int n) {
    int *ptr = &n;
    return (*ptr * (*ptr + 1)) / 2;
}
```

Terminal

```
Enter the number: 9.
The sum of 9 natural numbers is 45.
```

Teacher's Signature

Date

Expt. No. 68 Page No.

Objective: Swap two numbers using pointers.

C program

```
#Include <stdio.h>
Void swap(int *, int *);

Int main() {
    Int a,b;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);

    swap(&a, &b);
    printf("The values after swapping a=%d, b=%d",a,b);
    return 0;
}
void swap(int *a, int *b) {
    *a = *a ^ *b;
    *b = *a ^ *b;
    *a = *a ^ *b;
}
```

Terminal

Enter two numbers: 3 4
The values after swapping a=4, b=3

Teacher's Signature

Objective: Display values of a and b using functions.

C program

```
#include <stdio.h>
void display (int );
int main () {
    int n;
    printf ("Enter the number: ");
    scanf ("%d", &n);
    display (n);

    return 0;
}
void display (int a) {printf ("%d", a); }
```

Terminal

```
Enter the number: 5
 5
```

Expt. No. 70 Page No.

Objective : Display (i) 1-D array (ii) 2-D array using pointer.

C program

```
#include <stdio.h>
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int *ptr = arr;

    for(int i=0; i<n; i++) printf("%d ", *ptr++);
    printf("\n");

    int a[3][3] = {{1,2,3},{4,5,6},{7,8,9}};

    for(int i=0; i<3; i++) {
        for(int j=0; j<n; j++) printf("%d", (*a[i])++);
        printf("\n");
    }
    return 0;
}
```

Terminal

```
1 2 3 4 5
1 2 3
4 5 6
7 8 9
```

Teacher's Signature

Expt. No. 71 Page No.

Objective: Print a 2D array using pointers.

Theory: When we define a pointer to pointer, first pointer stores the address of variable and second pointer stores the address of first pointer.

C program:

```
#include <stdio.h>
int main() {
    int a[3][3] = {{1,2,3}, {4,5,6}, {7,8,9}}

    for (int i=0; i<3; i++) {
        for (int j=0; j<3; j++) printf("%d", *(*(a+i)+j));
        printf("\n");
    }

    return 0;
}
```

Terminal

```
1 2 3
4 5 6
7 8 9
```

Teacher's Signature

Expt. No. 72 Page No.

Objective: Print the address and entered integer with hexadecimal value.

Theory: To print address in hexadecimal format, we use %p specifier.

C program:

```
#Include <stdio.h>
int main() {
    int n;
    printf("Enter the number: ");
    scanf("%d", &n);

    printf("The address of %d is %p", n, &n);

    return 0;
}
```

Terminal

```
Enter the number: 7
The address of 7 is 000000000065FE1C
```

Teacher's Signature

Expt No. 13 Page No.

**Objective:** Create a structure which stores and display the player's name and jersey number.

**Theory:** Structure is a user-defined data type in C language which allows us to combine data of different types together.

**C program:**

```
#Include <stdio.h>
int main() {
typedef struct {
    char name[30]; int jersey;
} player;

int main() {
    Player ;

    Printf("Enter the name: ");
    scanf("%[^\n]*c", name);
                      player

    printf("Enter the jersey number: ");
    scanf("%d", jersey);
                player

    printf("player %s has jersey number %d", .name, .jersey);
                                              player  player

    return 0;
}
```

Teacher's Signature _____

Date _____

Expt. No. 74 Page No. _____

objective: Create an array which stores and display the names and jersey number of players.

C program

```
#include <stdio.h>
struct Player {
    char name[30]; int jersey;
};

int main() {
struct Player arr[i];

    for(int i=0; i<11; i++){
        printf("Player Name: ");
        scanf("%[^\n]*c", &arr[i].name);
        printf("Jersey Number: ");
        scanf("%d", &arr[i].jersey);
    }

    for(int i=0; i<11; i++)
        printf("%s has jersey number %d\n", arr[i].name arr[i].jersey);

    }
    return 0;
}
```

Teacher's Signature _____

Expt. No. 15 Page No.

objective : Create an alias name of the structure

Theory : Typedef is used to create aliases.

C program :

```
#include<stdio.h>

typedef struct {
        char name[30];
        int jersey;
}player;

int main() {
    player one;
    printf(" Enter the name: ");
    scanf("%s", &one.name);

    printf("Enter the jersey: ");
    scanf("%d", &one.jersey);

    printf("%s :wears jersey number %d", one.name, one.jersey);

    return 0;
}
```

Terminal

```
Enter the name: Prince
Enter the jersey : 11
Prince wears jersey number 11
```

Teacher's Signature