**Linux:**

* Linux was created by Linus Torvalds in 1991.
* Linux is a UNIX-based operating system originally developed as for Intel-compatible PC's.
* It is now available for most types of hardware platforms, ranging from PDAs (Personal Digital Assistant) to mainframes.
* Linux is an open source operating system. Anyone can access its coding from Internet and edit its coding.
* There is a very large collection of free software available for Linux. (Graphical environments (GUIs), office applications, developers' tools, system utilities, business applications, document publishing tools, network client and server applications etc.)

**Why Linux:**

* Linux is free.
  + Can view and edit the source code of OS
* It is fully customizable.
* Most Important Feature is Stability
  + Bugs are fixed very quickly.
* Linux has better security structure.
* Linux is a multitasking and multiuser operating system.
* High Portability
  + Easy to port in new H/W Platform
  + Written in C which is highly portable

**Command for compiling and running C program on Linux:**

* In Linux, GCC compiler is installed. The command for compiling C program is gcc.
* If a C file name is “ex.c”. For compiling we have to type

**gcc ex.c**

* After compiling the program, the program is ready to run. After compiling “a.out” executable file is created. For running we have to type

**./a.out**

* Another command for compiling the program “hello.c”.

**gcc -o hello hello.c**

This command will invoke the GCC compiler to compile the file hello.c and output (-o) the result to an executable called hello.

**List of Programs**

1. C program for printing a line.
2. C program for printing multiple lines.
3. C program for printing the value of any variable.
4. C program for adding two numbers.
5. C program for adding two numbers given by user.
6. C program for dividing two numbers given by user.
7. C program for calculating the area of a circle of given radius.
8. C program for comparing two numbers.
9. Write a C program for finding out given number is even or odd.
10. Write a C program for finding out Profit and Loss in sale, if the cost price and selling price of product is given.
11. C program for finding out maximum number from three numbers.
12. C program to find out division in the exam according to percentage.
13. C program for checking eligibility in the exam.
14. C program for Arithmetic operation like addition, subtraction, multiplication, division using ‘switch-case’.
15. Write a C program for finding out given character is vowel or consonant.
16. Write a C program for printing numbers 0-9 in words.
17. Write a C program for printing the names of days (1 for Sunday, 2 for Monday, . . . , 7 for Saturday).
18. C program for printing the table of any number.
19. **to 30.** C program for printing Patterns.
20. C program for calculating the area of circle using function.
21. C program for calculating area of triangle, if sides of triangle are given.
22. C program for checking given year is leap year or not.
23. C program for finding out division in the class according to percentage.
24. C program for checking given number is prime or not.
25. C program for printing all prime numbers from 1 to 100.
26. C program for calculating sum of digits of given number.
27. C program for reversing given number.
28. C program for checking given number is palindrome or not.
29. C program for checking given number is octal number or not.
30. C program for calculating factorial of given number.
31. C program for printing n terms of Fibonacci series.
32. C program for printing nth term of Fibonacci series.
33. C program for calculating GCD (Greatest Common Divisor) of two numbers.
34. C program for calculating LCM (Least Common Multiple) of two numbers.
35. C program for calculating sum of n terms of series 1!+2!+3!+...+n!.
36. C program for calculating series 1/1! + 2/2! + 3/3!+ . . . +n/n!
37. C program for calculating series 1^2 + 2^2 + 3^2+ . . . +n^2!
38. C program for calculating series 1^2 - 2^2 + 3^2- . . . +-n^2!
39. C program for calculating series 2^2 + 4^2+ . . . +n^2!
40. C program for calculating series 1^2 - 3^2+ . . . +-n^2!
41. C program for converting decimal number to binary number.
42. C program for converting decimal number to octal number.
43. C program for converting Decimal number to Hexadecimal number.
44. C program for checking given number is Armstrong number or not.
45. Write a C program for finding out maximum number in an array.
46. C program for finding out average of 10 numbers.
47. C program for transpose of a matrix.
48. C program for addition of two matrices A and B.
49. Write a Program in C for multiplication of two matrices A and B.
50. Write a C program for swapping two numbers using call by values function.
51. Write a C program for swapping two numbers using call by reference function.
52. C program for calculating the length of any string using library function (strlen).
53. C program for calculating the length of any string without using library function.
54. C program for checking given string is Palindrome or not.
55. C program for finding out Factorial of a number using recursion.
56. Write a C program for printing the contents of a file on screen.
57. C program to copy the contents of one file to another.
58. Create a structure for book(name, price, pages).
59. Create a structure to specify data on students given below:

Roll number, Name, Department, Course, Year of joining.

Assume that there are not more than 100 students in the college.

Write a function to print names of all students who joined in a particular year.

Write a function to print the data of a student whose roll number is given.

1. **C program for printing a line.**

#include<stdio.h>

void main()

{

printf(“Hindustan Institute of Technology and Management”);

}

**Output:**

Hindustan Institute of Technology and Management

1. **C program for printing multiple lines. For this, we use a character ‘\n’ which creates new line. The following program is for printing.**

**--------------------**

**I love my India.**

**--------------------**

#include<stdio.h>

void main()

{

printf(“--------------------”);

printf(“I love my India.”);

printf(“--------------------”);

}

**Output:**

--------------------

I love my India.

--------------------

1. **C program for printing the value of any variable. For this, we use the format of that variable like int- %d, float- %f, char- %c.**

#include<stdio.h>

void main()

{

int i; float f; char c; i=8; f=9.56; c=’a’;

printf(“The value of i is %d.”, i);

printf(“\n The value of f is %f.”, f);

printf(“\n The value of c is %c.”, c);

}

**Output:**

The value of i is 8.

The value of f is 9.560000.

The value of c is a.

1. **C program for adding two numbers.**

#include<stdio.h>

void main()

{

int a, b, c; a=12; b=25; c=a+b;

printf(“The addition of %d and %d is %d.”, a, b, c);

}

**Output:**

The addition of 12 and 25 is 37.

1. **C program for adding two numbers given by user. For this we use ‘scanf’ function. In ‘scanf’ function we define the format of the variable and pass address of that variable.**

#include<stdio.h>

void main()

{

int a, b, c;

printf(“Enter one number : ”);

scanf(“%d”,&a); // &a- address of a.

printf(“\n Enter second number: ”);

scanf(“%d”,&b); //&b- address of b.

c=a+b;

printf(“\n The addition is %d .”,c);

}

**Output:**

Enter one number: 51

Enter second number: 25

The addition is 76.

1. **C program for dividing two numbers given by user.**

#include<stdio.h>

void main()

{

float a, b, c;

printf(“Enter one number: “); scanf(“%f”,&a);

printf(“\n Enter second number: “); scanf(“%f”, &b);

c=a/b;

printf(“\n The division is %f “,c);

}

**Output:**

Enter one number: 1

Enter second number: 2

The addition is 0.500000

1. **C program for calculating the area of a circle of given radius. (area= π r2).**

#include<stdio.h>

void main()

{

float radius, area;

printf(“Enter radius of circle: “); scanf(“%f”,&radius); area=3.14\*radius\*radius;

printf(“\n The area of circle is %f”, area);

}

**Output:**

Enter radius of circle: 3

The area of circle is 28.260000

1. **C program for comparing two numbers. For comparing, we use relational operators like (<, <=, >, >=, ==, !=) and if-else clause.**

#include<stdio.h>

void main()

{

int a, b;

printf(“Enter two numbers: ”); scanf(“%d%d”,&a, &b); if(a==b)

printf(“\n Both numbers are same”); else if(a>b)

printf(“\n %d is greater than %d.”, a, b);

else

printf(“\n %d is greater than %d”, b, a);

}

**Output:**

Enter two numbers: 5 3

5 is greater than 3

1. **Write a C program for finding out given number is even or odd.**

#include<stdio.h>

void main()

{

int num;

printf(“Enter one number:”); scanf(“%d”,&num); if(num%2==0)

printf(“%d is even.”,num);

else

printf(“%d is odd.”,num);

}

**Output:**

Enter one number: 24

1. even.
2. **Write a C program for finding out Profit and Loss in sale, if the cost price and selling price of product is given.**

#include<stdio.h>

void main()

{

int CP, SP;

printf(“Enter cost price of product:”); scanf(“%d”,&CP);

printf(“\nEnter selling price of product:”); scanf(“%d”,&SP);

if(SP>CP)

{

printf(“\nProfit is %d”,SP-CP);

}

else

printf(“\nLoss is %d”,CP-SP);

}

**Output:**

Enter cost price of product: 20

Enter selling price of product: 22

Profit is 2

1. **C program for finding out maximum number from three numbers. In this program, we will use nested if.**

#include<stdio.h>

void main()

{

int a, b, c;

printf(“Enter three numbers: ”); scanf(“%d%d%d”,&a, &b, &c); if(a>b)

{

if(a>c)

printf(“\n %d is maximum.”, a);

else

printf(“\n %d is maximum.”, c);

}

else if (b>c)

printf(“\n %d is maximum.”, b);

else

printf(“\n %d is maximum.”,c);

}

**Output:**

Enter three numbers: 3 6 8 8 is maximum.

1. **C program to find out division in the exam according to percentage. If percentage is greater than 60 then First Division; if percentage is greater than 45 then Second Division; if percentage is greater than 33 then Third Division; otherwise Fail.**

#include<stdio.h>

void main()

{

float percentage;

printf(“Enter your percentage scored in the exam: ”);

scanf(“%f”,&percentage);

if(percentage>=60)

printf(“\n First Division.”); else if(percentage>=45)

printf(“\n Second Division.”); else if(percentage>=33)

printf(“\n Third Division.”);

else

printf(“\n Fail”);

}

**Output:**

Enter your percentage scored in the exam: 35

Third Division.

1. **C program for checking eligibility in the exam by using logical operators like (&&, ||, !).**

#include<stdio.h>

void main()

{

int age;

float percentage;

printf(“Enter your age:”); scanf(“%d”,&age);

printf(“\n Enter your percentage of degree:”); scanf(“%f”,&percentage);

if(age<=30 && percentage>=70)

printf(“\n You are eligible in the exam.”);

else

printf(“\n You are not eligible in the exam.);

}

**Output:**

Enter your age: 25

Enter your percentage of degree: 71.50

You are eligible in the exam.

**switch-case:-** We use switch-case when we want to do more than one operation by giving choice.

1. **C program for Arithmetic operation like addition, subtraction, multiplication, division using ‘switch-case’.**

#include<stdio.h>

void main()

{

int n;

float a, b, add, sub, mul, div;

printf(“Enter two numbers:”);

scanf(“%f%f”, &a, &b);

printf(“\n Enter choice.”);

printf(“\n 1. Addition \n 2. Subtraction \n 3. Multiplication \n 4. Division \n”); scanf(“%d”,&n);

switch(n)

{

case 1: add=a+b;

printf(“\n The addition is %f”,add);

break;

case 2: sub=a-b;

printf(“\n The subtraction is %f”,sub);

break;

case 3: mul=a\*b;

printf(“\n The multiplication is %f”,mul); break;

case 4: div=a/b;

printf(“\n The division is %f”,div); break;

default: printf(“\n You entered wrong choice.”);

}

}

**Output:**

Enter two numbers: 15 4

Enter choice.

1. Addition
2. Subtraction
3. Multiplication
4. Division

3

The multiplication is 60.000000

1. **Write a C program for finding out given character is vowel or consonant.**

#include<stdio.h>

void main()

{

char ch;

printf(“Enter any alphabet:”);

scanf(“%d”,&ch);

switch(ch)

{

case ‘a’:

case ‘A’:

case ‘e’:

case ‘E’:

case ‘i':

case ‘I’:

case ‘o’:

case ‘O’:

case ‘u’:

case ‘U’: printf(“vowel”); break;

default: printf(“consonant”);

}

}

**Output:**

Enter any alphabet:

E

vowel

1. **Write a C program for printing numbers 0-9 in words.**

#include<stdio.h>

void main()

{

int n;

printf(“Enter one number from 0 to 9”); scanf(“%d”,&n);

switch(n)

{

case 0: printf(“zero”); break; case 1: printf(“one”); break; case 2: printf(“two”); break; case 3: printf(“three”); break; case 4: printf(“four”); break; case 5: printf(“five”); break; case 6: printf(“six”); break; case 7: printf(“seven”); break; case 8: printf(“eight”); break; case 9: printf(“nine”); break; default: printf(“wrong input”);

}

}

**Output:**

Enter one number from 0 to 9

6

six

1. **Write a C program for printing the names of days (1 for Sunday, 2 for Monday, . . . , 7 for Saturday).**

#include<stdio.h>

void main()

{

int n;

printf(“Enter one number from 1 to 7”);

scanf(“%d”,&n);

switch(n)

{

case 1: printf(“Sunday”);

break;

case 2: printf(“Monday”);

break;

case 3: printf(“Tuesday”);

break;

case 4: printf(“Wednesday”);

break;

case 5: printf(“Thursday”);

break;

case 6: printf(“Friday”);

break;

case 7: printf(“Saturday”);

break;

default: printf(“wrong input”);

}

}

**Output:**

Enter one number from 1 to 7

4

Wednesday

**Loop:-** We use loop in programming, when we want to execute same line or lines many times. There arethree loops in C (while, do-while, and for).

**while**(*CONDITION*)

{

}

**do**

{

}**while**(*CONDITION*);

**for**(initialization ;condition ; increment or decrement)

{

}

1. **C program for printing the table of any number using ‘while’ loop.**

#include<stdio.h>

void main()

{

int n, i, t;

printf(“Enter one number: ”);

scanf(“%d”,&n);

i=1;

printf(“\n The table of %d number is \n”,n);

while(i<=10)

{

t=n \* i;

printf(“%d\t”,t);

i++;

}

}

**Output:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enter one number: 3 | | |  |  |  |  |  |  |  |
| The table of 3 is | |  |  |  |  |  |  |  |  |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |

1. **C program for generating pattern**

**1**

**12**

**123**

**1234**

**12345**

#include<stdio.h>

void main()

{  
 int i, j;

for(i=1;i<=5;i++)

{

for(j=1;j<=i;j++)

{

printf(“%d”,j);

}

printf(“\n”);

}

}

1. **C program for generating pattern**

**1**

**22**

**333**

**4444**

**55555**

#include<stdio.h>

void main()

{  
 int i, j;

for(i=1;i<=5;i++)

{

for(j=1;j<=i;j++)

{

printf(“%d”,i);

}

printf(“\n”);

}

}

1. **C program for generating pattern**

**12345**

**1234**

**123**

**12**

**1**

#include<stdio.h>

void main()

{  
 int i, j;

for(i=1;i<=5;i++)

{

for(j=1;j<=6-i;j++)

{

printf(“%d”,j);

}

printf(“\n”);

}

}

1. **C program for generating pattern**

**12345**

**2345**

**345**

**45**

**5**

#include<stdio.h>

void main()

{  
 int i, j;

for(i=1;i<=5;i++)

{

for(j=i;j<=5;j++)

{

printf(“%d”,j);

}

printf(“\n”);

}

}

1. **C program for generating pattern**

**5**

**54**

**543**

**5432**

**54321**

#include<stdio.h>

void main()

{  
 int i, j;

for(i=1;i<=5;i++)

{

for(j=5;j>=6-i;j--)

{

printf(“%d”,j);

}

printf(“\n”);

}

}

1. **C program for generating pattern**

**54321**

**5432**

**543**

**54**

**5**

#include<stdio.h>

void main()

{  
 int i, j;

for(i=1;i<=5;i++)

{

for(j=5;j>=i;j--)

{

printf(“%d”,j);

}

printf(“\n”);

}

}

1. **C program for generating pattern**

**54321**

**4321**

**321**

**21**

**1**

#include<stdio.h>

void main()

{  
 int i, j;

for(i=1;i<=5;i++)

{

for(j=6-i;j>=1;j--)

{

printf(“%d”,j);

}

printf(“\n”);

}

}

1. **C program for generating pattern**

**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\*\*\***

#include<stdio.h>

void main()

{  
 int i, j;

for(i=1;i<=5;i++)

{

for(j=1;j<=i;j++)

{

printf(“\*”);

}

printf(“\n”);

}

}

1. **C program for generating pattern**

**123454321**

**1234 4321**

**123 321**

**12 21**

**1 1**

#include<stdio.h>

void main()

{

int i,j;

for(i=1;i<=5;i++)

{

for(j=1;j<=6-i;j++)

printf("%d",j);

for(j=1;j<=2\*i-3;j++)

printf(" ");

for(j=6-i;j>=i;j--)

{

if(i!=1||j!=5)

printf("%d",j);

}

printf("\n");

}

}

1. **C program for generating pattern**

**1**

**121**

**12321**

**1234321**

**123454321**

#include<stdio.h>

void main()

{

int i, j;

for(i=1;i<=5;i++)

{

for(j=4;j>=i;j--)

{

printf(" ");

}

for(j=1;j<=i;j++)

{

printf("%d",j);

}

for(j=i-1;j>=1;j--)

{

printf("%d",j);

}

printf("\n");

}

}

1. **C program for generating pattern**

**1**

**123**

**12345**

**1234567**

**123456789**

#include<stdio.h>

void main()

{

int i, j;

for(i=1;i<=5;i++)

{

for(j=4;j>=i;j--)

{

printf(" ");

}

for(j=1;j<=2\*i-1;j++)

{

printf("%d",j);

}

printf("\n");

}

}

1. **C program for generating pattern**

**ABCDEFGFEDCBA**

**ABCDEF FEDCBA**

**ABCDE EDCBA**

**ABCD DCBA**

**ABC CBA**

**AB BA**

**A A**

#include<stdio.h>

void main()

{

int i,j;

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

{

for(j=1;j<=8-i;j++)

{

printf("%c",64+j);

}

for(j=1;j<=2\*i-3;j++)

{

printf(" ");

}

for(j=1;j<=8-i;j++)

{

if(i!=1 || j!=1)

printf("%c",73-i-j);

}

printf("\n");

}

}

**User-Defined Function:-** There are times when certain type of operations or calculations are repeated atmany points throughout a program. For instance, we might use the factorial of a number at several points in the program. In such situations, we may repeat the program statements wherever they are needed.

Another approach is to design a function that can be called and used whenever required. This saves both time and space.

1. **C program for calculating the area of circle (area= π r2).**

#include<stdio.h>

float C\_Area(float); //function declaration.

void main()

{

float r, area;

printf("Enter radius of circle ");

scanf("%f",&r);

area=C\_Area(r); //function calling

printf("Area of circle is %f",area);

}

float C\_Area(float r) //function definition

{

float area;

area=(22/7.0)\*r\*r;

return(area);

}

1. **C program for calculating area of triangle, if sides of triangle are given.**

**Concept:-** A triangle is possible if sum of two sides is greater than third side i.e. ( if sides of triangle are a, b, c; then (a+b)>c and (b+c)>a and (c+a)>b).

Formula of calculating area of triangle if we know all three sides.

s=(a+b+c)/2

area= (s \* (s-a) \* (s-b) \* (s-c) )1/2

#include<stdio.h>

#include<math.h>

void T\_Area(float a, float b, float c)

{

float s, area;

if((a+b)>c && (b+c)>a && (c+a)>b)

{

s=(a+b+c)/2;

area=sqrt(s\*(s-a)\*(s-b)\*(s-c));

printf("Area of Triangle is %f",area);

}

else

printf("Triangle is not possible");

}

void main()

{

float a, b, c;

printf("Enter sides of triangle ");

scanf("%f%f%f",&a, &b, &c);

T\_Area(a,b,c);

}

1. **C program for checking given year is leap year or not.**

**Concept:-** If any year is divisible by 400, then that year is Leap Year. If that year is not divisible by 400, but divisible by 4 and not divisible by 100, that year is Leap year.

#include<stdio.h>

void Leap\_Year(int);

void main()

{

int year;

printf("Enter one year ");

scanf("%d",&year);

Leap\_Year(year);

}

void Leap\_Year(int year)

{

if(year%400==0 || (year%4==0 && year%100!=0))

{

printf("%d is leap year",year);

}

else

{

printf("%d is not leap year", year);

}

}

1. **C program for finding out division in the class according to percentage. If percentage is greater than 60 then First Division; if percentage is greater than 45 then Second Division; if percentage is greater than 30 then Third Division; otherwise Fail.**

#include<stdio.h>

void Division(int);

void main()

{

int per;

printf("Enter your percentage ");

scanf("%d",&per);

Division(per);

}

void Division(int per)

{

if(per>=60)

printf("First Division");

else if(per>=45)

printf("Second Division");

else if(per>=30)

printf("Third Division");

else

printf("Fail");

}

1. **C program for checking given number is prime or not.**

**Concept:-** Prime numbers are those numbers which are divisible by only 1 or itself. For this we have to divide any number(n) by 2 to n/2. If that number is divisible by any number from 2 to n/2, then that number is not prime number, otherwise that number is prime number.

#include<stdio.h>

void Prime(int);

void main()

{

int n;

printf("Enter one number greater than one ");

scanf("%d",&n);

Prime(n);

}

void Prime(int n)

{

int i;

i=2;

while(i<=n/2)

{

if(n%i==0)

{

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

break;

}

i++;

}

if(i>n/2)

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

}

1. **C program for printing all prime numbers from 1 to 100.**

#include<stdio.h>

void All\_Prime();

void main()

{

All\_Prime();

}

void All\_Prime()

{

int i,x;

printf("All prime numbers from 1 to 100\n");

for(x=2;x<=100;x++)

{

i=2;

while(i<=x/2)

{

if(x%i==0)

{

break;

}

i++;

}

if(i>x/2)

printf("%d ",x);

}

}

1. **C program for calculating sum of digits of given number. (eg. If number is 268, then sum of digits is 2+6+8= 16).**

**Concept:-** If any number is divided by 10, then last digit of that number will be remainder; and remaining digits will be quotient. By applying this process repeatedly we can separate digits from number and calculate sum of digits of that number by using formula (***sum= sum +r*** ), where initial value of *sum* is zero and *r* is the remainder. For division we use operator (**/** ) and for remainder we use modulus operator (%).

#include<stdio.h>

void Sum\_of\_Digits(int);

void main()

{

int n;

printf("Enter one number ");

scanf("%d",&n);

Sum\_of\_Digits(n);

}

void Sum\_of\_Digits(int n)

{

int x, r, sum=0;

x=n;

while(x!=0)

{

r=x%10;

sum=sum+r;

x=x/10;

}

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

}

1. **C program for reversing given number. (eg. Reverse of number( 473) is 374).**

**Concept:-** we have to separate digits from number from the concept of sum of digits program. After separate one digit we apply the formula (***sum= sum\*10 + r***), where initial value of *sum* is zero, and ***r*** is the remainder. This process is applied repeatedly.

#include<stdio.h>

void Reverse(int);

void main()

{

int n;

printf("Enter one number ");

scanf("%d",&n);

Reverse(n);

}

void Reverse(int n)

{

int x, r, sum=0;

x=n;

while(x!=0)

{

r=x%10;

sum=sum\*10+r;

x=x/10;

}

printf("Reverse of %d is %d",n, sum);

}

1. **C program for checking given number is palindrome or not. ( If the reverse of any number is same as original number, then that number is called palindrome number. Eg. 121, 111, 12321 etc.)**

**Concept:-** we have to find out reverse of given number. If reversed number is same as original number, then write Palindrome; otherwise Not Palindrome.

#include<stdio.h>

void Palindrome(int);

void main()

{

int n;

printf("Enter one number ");

scanf("%d",&n);

Palindrome(n);

}

void Palindrome(int n)

{

int x, r, sum=0;

x=n;

while(x!=0)

{

r=x%10;

sum=sum\*10+r;

x=x/10;

}

if(n==sum)

printf("%d is palindrome number",n);

else

printf("%d is not palindrome number",n);

}

1. **C program for checking given number is octal number or not.**

**Concept:-** In Octal number digits (8 and 9) are not present, only 0 to 7 are present. Therefore we have to separate digits from number and check that digit is 8 or 9. If any digit is 8 or 9, that number cannot be octal number; otherwise that number is octal number.

#include<stdio.h>

void Octal(int);

void main()

{

int n;

printf("Enter one number ");

scanf("%d",&n);

Octal(n);

}

void Octal(int n)

{

int x, r;

x=n;

while(x!=0)

{

r=x%10;

if(r==8||r==9)

{

printf("%d is not octal number",n);

break;

}

x=x/10;

}

if(x==0)

printf("%d is octal number",n);

}

1. **C program for calculating factorial of given integer number. (eg. Factorial of 5 is 1\*2\*3\*4\*5= 120).**

**Concept:-** we use formula (***fact= fact \* i***) where initial value of *fact* is 1, and initial value of *i* is 1 or 2. We use this formula repeatedly from (*i*= 2 to *n*); where *n* is a given integer number.

#include<stdio.h>

long Factorial(int n)

{

long fact=1;

int i;

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

{

fact=fact\*i;

}

return(fact);

}

void main()

{

int n;

long fact;

printf("Enter one number for calculating factorial ");

scanf("%d",&n);

fact=Factorial(n);

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

}

1. **C program for printing *n* terms of Fibonacci series (0, 1, 1, 2, 3, 5, 8, 13, 21, . . . ).**

**Concept:-** First take two integer variable *a*, *b* with initial value of *a*=0, *b*=1. Now run one loop from (*i*= 1 to *n*) where *n* is number of terms. In every iteration, print the value of variable *a*; and calculate addition of *a* and *b* and store that value into third variable *c*; now update value of *a* and *b* (*a=b*, *b=c*).

#include<stdio.h>

void Fibo(int n)

{

int a=0, b=1, c, i;

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

{

printf("%d ",a);

c=a+b;

a=b;

b=c;

}

}

void main()

{

int n;

printf("How many terms do you want in Fibonacci series ");

scanf("%d",&n);

Fibo(n);

}

1. **C program for printing *n*th term of Fibonacci series.**

**Concept:-** in this program, printing will be outside the loop and loop will be executed upto (*n*-1).

#include<stdio.h>

void Fibo(int n)

{

int a=0, b=1, c, i;

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

{

c=a+b;

a=b;

b=c;

}

printf("%d term is %d", n, a);

}

void main()

{

int n;

printf("Which term do you want in Fibonacci series ");

scanf("%d",&n);

Fibo(n);

}

1. **C program for calculating GCD (Greatest Common Divisor) of two numbers.**

**Concept:-** The GCD of two number is that greatest number which can divide completely both numbers separately. (eg. GCD of two numbers *a*=27 and *b*=18 is 9).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 18 (*b*) | 27 (*a*) | 1 |  |  |
|  | 18 |  |  |  |
|  | 9 (*r*) |  |  |  |
|  | 9 (*b*) | 18 (*a*) | 2 |  |
|  |  | 18 |  |  |
|  |  | 0 |  |  |
|  |  | 0 (*b*) | **9 (*a*)** |  |
|  |  |  | GCD is 9 |  |

#include<stdio.h>

int GCD(int a, int b)

{

int r;

while(b!=0)

{

r=a%b;

a=b;

b=r;

}

return(a);

}

void main()

{

int gcd, a, b;

printf("Enter two numbers for calculating GCD ");

scanf("%d%d",&a,&b);

gcd=GCD(a,b);

printf("GCD is %d",gcd);

}

1. **C program for calculating LCM (Least Common Multiple) of two numbers.**

**Concept:-** LCM of two numbers is that least number which can be divisible by both given numbers separately. (eg. LCM of two numbers (a= 27, b= 18) is 54). The formula for calculating LCM is [**(a\*b)/ GCD(a,b)**].

#include<stdio.h>

int GCD(int a, int b)

{

int r;

while(b!=0)

{

r=a%b;

a=b;

b=r;

}

return(a);

}

void LCM(int a, int b)

{

int lcm;

lcm=(a\*b)/GCD(a,b);

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

}

void main()

{

int a, b;

printf("Enter two numbers for calculating LCM ");

scanf("%d%d",&a,&b);

LCM(a,b);

}

1. **C program for calculating sum of n terms of series 1!+2!+3!+...+n!.**

#include<stdio.h>

long Factorial(int n)

{

long fact=1;

int i;

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

{

fact=fact\*i;

}

return(fact);

}

void Fact\_Series(int n)

{

long sum=0;

int i;

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

{

sum=sum+Factorial(i);

}

printf("Sum of series is %ld",sum);

}

void main()

{

int n;

printf("Enter value of n for series 1!+2!+...+n! ");

scanf("%d",&n);

Fact\_Series(n);

}

1. **C program for calculating series 1/1! + 2/2! + 3/3!+ . . . +n/n!**

#include<stdio.h>

void Factorial\_Series(int);

long Factorial(int);

void main()

{

int n;

printf("Enter value of n for series 1/1!+2/2!+...+n/n! ");

scanf("%d",&n);

Factorial\_Series(n);

}

void Factorial\_Series(int n)

{

float sum=0;

int i;

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

{

sum=sum+i\*1.0/Factorial(i);

}

printf("\n Sum of series is %f",sum);

}

long Factorial(int n)

{

long fact=1;

int i;

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

{

fact=fact\*i;

}

return(fact);

}

1. **C program for calculating series 1^2 + 2^2 + 3^2+ . . . +n^2!**

#include<stdio.h>

void Series(int);

void main()

{

int n;

printf("Enter value of n for series 1^2+2^2+...+n^2 ");

scanf("%d",&n);

Series(n);

}

void Series(int n)

{

int i, sum=0;

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

{

sum=sum+i\*i;

}

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

}

1. **C program for calculating series 1^2 - 2^2 + 3^2- . . . +-n^2!**

#include<stdio.h>

void Series(int);

void main()

{

int n;

printf("Enter value of n for series 1^2-2^2+...+-n^2 ");

scanf("%d",&n);

Series(n);

}

void Series(int n)

{

int i, sum=0;

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

{

if(i%2!=0)

{

sum=sum+i\*i;

}

else

{

sum=sum-i\*i;

}

}

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

}

1. **C program for calculating series 2^2 + 4^2+ . . . +n^2!**

#include<stdio.h>

void Series(int);

void main()

{

int n;

printf("Enter value of n for series 2^2+4^2...+n^2 ");

scanf("%d",&n);

Series(n);

}

void Series(int n)

{

int i, sum=0, a=2;

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

{

sum=sum+a\*a;

a=a+2;

}

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

}

1. **C program for calculating series 1^2 - 3^2+ . . . +-n^2!**

#include<stdio.h>

void Series(int);

void main()

{

int n;

printf("Enter value of n for series 1^2-3^2...+-n^2 ");

scanf("%d",&n);

Series(n);

}

void Series(int n)

{

int i, sum=0, a=1;

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

{

if(i%2!=0)

{

sum=sum+a\*a;

}

else

{

sum=sum-a\*a;

}

a=a+2;

}

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

}

1. **C program for converting decimal number to binary number. (eg. (12)10= (1100)2).**

**Concept:-** For converting Decimal number into Binary number we have to divide decimal number by 2 and store remainder into an array. Again apply this process on quotient until quotient be zero.

#include<stdio.h>

void Deci\_Bin(int);

void main()

{

int n;

printf("Enter one decimal number ");

scanf("%d",&n);

Deci\_Bin(n);

}

void Deci\_Bin(int n)

{

int A[20], i, r, x, len;

x=n;

i=0;

while(x!=0)

{

r=x%2;

A[i]=r;

i++;

x=x/2;

}

len=i;

printf("\nBinary Equivalent is \n");

for(i=len-1;i>=0;i--)

{

printf("%d",A[i]);

}

}

1. **C program for converting decimal number to octal number. (eg. (123)10 = (173)8).**

**Concept:-** For converting Decimal number into Octal number we have to divide decimal number by 8 and store remainder into an array. Again apply this process on quotient until quotient be zero.

#include<stdio.h>

void Deci\_Oct(int);

void main()

{

int n;

printf("Enter one decimal number ");

scanf("%d",&n);

Deci\_Oct(n);

}

void Deci\_Oct(int n)

{

int A[20], i, r, x, len;

x=n;

i=0;

while(x!=0)

{

r=x%8;

A[i]=r;

i++;

x=x/8;

}

len=i;

printf("\nOctal Equivalent is \n");

for(i=len-1;i>=0;i--)

{

printf("%d",A[i]);

}

}

1. **C program for converting Decimal number to Hexadecimal number.**

**Concept:-** For converting Decimal number into Hexa-decimal number we have to divide decimal number by 16 and store remainder into an array. For remainder 10 we have to store ‘A’, for 11 store ‘B’, … , for 15 store ‘F’. Again apply this process on quotient until quotient be zero.

#include<stdio.h>

void Deci\_Hexa(int);

void main()

{

int n;

printf("Enter one decimal number ");

scanf("%d",&n);

Deci\_Hexa(n);

}

void Deci\_Hexa(int n)

{

char A[20];

int i, r, x, len;

x=n;

i=0;

while(x!=0)

{

r=x%16;

switch(r)

{

case 10: A[i]='A';

break;

case 11: A[i]='B';

break;

case 12: A[i]='C';

break;

case 13: A[i]='D';

break;

case 14: A[i]='E';

break;

case 15: A[i]='F';

break;

default: A[i]=r+48;

}

i++;

x=x/16;

}

len=i;

printf("\nHexadecimal Equivalent is \n");

for(i=len-1;i>=0;i--)

{

printf("%c",A[i]);

}

}

1. **C program for checking given number is Armstrong number or not. Ex. 153**

**153- (1^3+5^3+3^3=1+125+27=153)**

#include<stdio.h>

#include<math.h>

void Armstrong(int);

void main()

{

int n;

printf("Enter one number ");

scanf("%d",&n);

Armstrong(n);

}

void Armstrong(int n)

{

int x, digit, r, sum;

digit=0;

sum=0;

x=n;

while(x!=0)

{

digit++;

x=x/10;

}

x=n;

while(x!=0)

{

r=x%10;

sum=sum+pow(r,digit);

x=x/10;

}

if(n==sum)

{

printf("%d is armstrong number",n);

}

else

{

printf("%d is not armstrong number", n);

}

}

**Array:** An array is a collection of similar elements. These similar elements could be all ints, or all floats, or all chars etc.

* The array of characters is called a ‘string’.
* The array of ints or floats is called an array.

1. **Write a C program for finding out maximum number in an array.**

#include<stdio.h>

void main()

{

int A[20], i, n, max;

printf(“How many numbers are in array:”);

scanf(“%d”,&n);

printf(“\n Enter %d numbers: ”, n);

for(i=0;i<n;i++)

scanf(“%d”,&A[i]);

max=A[0];

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

{

if(max<A[i])

{

max=A[i];

}

}

printf(“\n Maximum number is %d”,max);

}

**Output:**

How many numbers are in array: 5

Enter 5 numbers: 2 5 15 20 9

Maximum number is 20

1. **C program for finding out average of 10 numbers.**

#include<stdio.h>

#include<conio.h>

void main()

{

int a[10], i;

float avg;

clrscr();

printf("Enter 10 numbers: ");

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

{

scanf("%d",&a[i]);

}

avg=a[0];

for(i=1;i<10;i++)

{

avg=avg+a[i];

}

avg=avg/10;

printf("\n The average is %f. ",avg);

getch();

}

**Output:**

Enter 10 numbers: 1 2 3 4 5 6 7 8 9 10

The average is 5.500000

1. **C program for transpose of a matrix.**

#include<stdio.h>

#define ROW 3

#define COL 3

void main()

{

int A[ROW][COL], B[COL][ROW], i, j;

printf(“\nEnter %d numbers for Matrix: ”, ROW\*COL);

for(i=0;i<ROW;i++)

{  
 for(j=0;j<COL;j++)

{

scanf(“%d”, &A[i][j]);

}

}

for(i=0;i<ROW;i++)

{

for(j=0;j<COL;j++)

{

B[j][i]=A[i][j];

}

}

printf(“\n Matrix A is \n”);

for(i=0;i<ROW;i++)

{

for(j=0;j<COL;j++)

{

printf(“%d ”, A[i][j]);

}

printf(“\n”);

}

printf(“\n Transpose of A matrix is \n”);

for(i=0;i<COL;i++)

{

for(j=0;j<ROW;j++)

{

printf(“%d ”, B[i][j]);

}

printf(“\n”);

}

}

**Output:**

Enter 9 numbers: 1 2 3 4 5 6 7 8 9

Matrix A is

1 2 3

4 5 6

7 8 9

Transpose of A matrix is

1 4 7

2 5 8

3 6 9

1. **C program for addition of two matrices A and B.**

#include<stdio.h>

#include<conio.h>

#define ROW 3

#define COL 3

void main()

{

int A[ROW][COL], B[ROW][COL], C[ROW][COL];

int i, j;

printf(“\nEnter %d numbers for 1st matrix”, ROW\*COL);

for(i=0;i<ROW;i++)

{  
 for(j=0;j<COL;j++)

{

scanf(“%d”, &A[i][j]);

}

}

printf(“\nEnter %d numbers for 2nd matrix”, ROW\*COL);

for(i=0;i<ROW;i++)

{  
 for(j=0;j<COL;j++)

{

scanf(“%d”, &B[i][j]);

}

}

for(i=0;i<ROW;i++)

{

for(j=0;j<COL;j++)

{

C[i][j]= A[i][j]+B[i][j];

}

}

printf(“\n Addition of matrix A and B is \n”);

for(i=0;i<ROW;i++)

{

for(j=0;j<COL;j++)

{

printf(“%d ”, C[i][j]);

}

printf(“\n”);

}

}

1. **Write a Program in C for multiplication of two matrices A and B.**

#include<stdio.h>

#define M 3 //rows of 1st matrix

#define N 2 // columns of 1st matrix and rows of 2nd matrix

#define P 3 //columns of 2nd matrix

void main()

{

int A[M][N], B[N][P], C[M][P];

int i, j, k, total;

printf(“\nEnter %d numbers for 1st matrix ”, M\*N);

for(i=0;i<M;i++)

{  
 for(j=0;j<N;j++)

{

scanf(“%d”, &A[i][j]);

}

}

printf(“\nEnter %d numbers for 2nd matrix”, N\*P);

for(i=0;i<N;i++)

{  
 for(j=0;j<P;j++)

{

scanf(“%d”, &B[i][j]);

}

}

for(i=0;i<M;i++)

{

for(j=0;j<P;j++)

{

total=0;

for(k=0;k<N;k++)

{

total= total + A[i][k] \* B[k][j];

}

C[i][j]=total;

}

}

printf(“\n Multiplication of matrix A and B is \n”);

for(i=0;i<M; i++)

{

for(j=0;j<P; j++)

{

printf(“%d ”, C[i][j]);

}

printf(“\n”);

}

}

**Output:**

Enter 6 numbers for 1st Matrix 1 2 3 4 5 6

Enter 6 numbers for 2nd Matrix 2 4 5 6 1 3

Multiplication of Matrix A and B is

14 6 11

30 16 27

46 26 43

1. **Write a C program for swapping two numbers using call by values function.**

#include<stdio.h>

void swap(int ,int);

void main()

{

int a,b ;

printf(“Enter two numbers:”);

scanf(“%d%d”,&a,&b);

printf("\n Before Swapping a=%d, b=%d", a, b);

swap(a,b);

}

void swap(int a, int b)

{

int c;

c=a;

a=b;

b=c;

printf("\n After Swapping a=%d, b= %d", a, b);

}

**Output:**

Enter two numbers: 2 4

Before Swapping a=2, b=4

After Swapping a=4, b=2

1. **Write a C program for swapping two numbers using call by reference function.**

#include<stdio.h>

void swap(int\*, int\*);

void main()

{

int a=2,b=3;

printf("\n Before Swapping a=%d, b=%d", a, b); swap(&a, &b);

printf(“\n After Swapping a=%d, b=%d”, a, b);

}

void swap(int \*p, int \*q)

{

int c; c=\*p; \*p=\*q; \*q=c;

}

**Output:**

Enter two numbers: 2 4

Before Swapping a=2, b=4

After Swapping a=4, b=2

**String:**

* A group of characters can be stored in a character array. Character arrays are called strings.
* A string constant is a one-dimensional array of characters terminated by a null (‘\0’).
* Initialization of string can be done by following ways.
  + char name[]= {‘H’, ‘E’, ‘L’, ‘L’, ‘O’, ‘\0’};
  + char name[]= “HELLO”; in this assignment ‘\0’ is not necessary. Compiler inserts the null character automatically.

1. **C program for calculating the length of any string using library function (strlen).**

#include<stdio.h>

#include<string.h>

void main()

{  
 char arr[]= “Hindustan”;

int len;

len=strlen(arr);

printf(“\nThe length of string is %d”, len);

}

**Output:**

The length of string is 9

1. **C program for calculating the length of any string without using library function.**

#include<stdio.h>

#include<string.h>

int String\_Length(char \*);

void main()

{

char s[]="shashank";

int len;

len=String\_Length(s);

printf("\n The length of string is %d", len);

}

int String\_Length(char \*p)

{

int l=0;

while(\*p!='\0')

{

l++;

p++;

}

return(l);

}

**Output:**

The length of string is 8

1. **C program for checking given string is Palindrome or not.**

#include<stdio.h>

#include<string.h>

void main()

{

char s[10], t[10];

printf("Enter one string: ");

gets(s);

strcpy(t, s);

strrev(s);

if(strcmp(s,t)==0)

printf("Palindrome");

else

printf("Not Plalindrome");

}

**Output:**

Enter one string: Madam

Palindrome

**Recursion:**

* In C, it is possible for the functions to call themselves.
* A function is called ‘recursive’ if a statement within the body of a function calls the same function.

1. **C program for finding out Factorial of a number using recursion.**

#include<stdio.h>

#include<conio.h>

int Factorial(int);

void main()

{  
 int a, fact;

clrscr();

printf(“Enter one number”);

scanf(“%d”, &a);

fact= Factorial(a);

printf(“\n The Factorial is %d.”, fact);

getch();

}

int Factorial(int x)

{  
 if(x==1)

return(1);

else

return(x\*Factorial(x-1));

}

1. **Write a C program for printing the contents of a file on screen.**

#include<stdio.h>

void main()

{

FILE \*fp;

char ch;

fp=fopen("sha.c","r");

if(fp==NULL)

{

printf("Cannot open source file");

exit(1);

}

while(1)

{

ch=fgetc(fp);

if(ch==EOF)

{

break;

}

printf("%c", ch);

}

fclose(fp);

}

1. **C program to copy the contents of one file to another.**

#include<stdio.h>

void main()

{

FILE \*fs, \*ft;

char ch;

fs=fopen("sha.c","r");

if(fs==NULL)

{

puts("Cannot open source file.");

exit(1);

}

ft=fopen("t.c","w");

if(ft==NULL)

{

puts("Cannot open target file.");

exit(1);

}

while(1)

{

ch=fgetc(fs);

if(ch==EOF)

{

break;

}

fputc(ch,ft);

}

fclose(fs);

fclose(ft);

}

**Structure:** Arrays can hold a number of pieces of information of the same data type. By structure we can store information of the different data types.

1. **Create a structure for book(name, price, pages).**

#include<stdio.h>

struct book

{  
 char name[10];

float price;

int pages;

};

void main()

{  
 struct book b1, b2;

printf(“Enter name, price, and pages of first book\n”);

scanf(“%s%f%d”, b1.name, &b1.price, &b1.pages);

printf(“Enter name, price, and pages of second book\n”);

scanf(“%s%f%d”, b2.name, &b2.price, &b2.pages);

printf(“\n The first book name is %s, price is %f, pages are %d”, b1.name, b1.price, b1.pages);

printf(“\n The second book name is %s, price is %f, pages are %d”, b2.name, b2.price, b2.pages);

}

1. **Create a structure to specify data on students given below:**

**Roll number, Name, Department, Course, Year of joining.**

**Assume that there are not more than 100 students in the college.**

**Write a function to print names of all students who joined in a particular year.**

**Write a function to print the data of a student whose roll number is given.**

#include<stdio.h>

void Print\_Name();

void Print\_Data();

struct student

{

int rn;

char name[20];

char dept[20];

char course[20];

int year;

}S[100];

void main()

{

int i;

for(i=0;i<100;i++)

{

printf("Enter roll no., name, department, course, joining year");

scanf("%d",&S[i].rn);

fflush(0);

gets(S[i].name);

fflush(0);

gets(S[i].dept);

fflush(0);

gets(S[i].course);

fflush(0);

scanf("%d",&S[i].year);

}

Print\_Name();

Print\_Data();

}

void Print\_Name()

{

int year, i;

printf("\n enter year");

scanf("%d",&year);

for(i=0;i<100;i++)

{

if(year==S[i].year)

{

printf("\n%s",S[i].name);

}

}

}

void Print\_Data()

{

int r, i;

printf("\n enter roll no.");

scanf("%d",&r);

for(i=0;i<100;i++)

{

if(r==S[i].rn)

{

printf("\n%s",S[i].name);

puts(S[i].dept);

puts(S[i].course);

printf("%d",S[i].year);

break;

}

}

}