Table of Contents

First c program to print Hello message

```
# include<stdio.h>

/*

This is our first c program

which is awesome!

*/

int main()

{

   printf("Hello I am learning C with Bsc Students");

   return 0;

}
```

Declaration and initialization of variables

```
#include <stdio.h>

/*

This is our first c program

which is awesome!

*/

int main()

{

   int a = 4.67;

   // int b = 8.5; // Not recommended because 8.5 is not an integer

   float b = 8.5;

   char c = 'u';

   int d = 45;

   int e = 45 + 4;

   printf("The value of a is %f \n", a);

   printf("The value of b is %f \n", b);

   printf("The value of c is %c \n", c);

   printf("Sum of a and d is %d \n", a - d);

   printf("Sum of a and d is %d \n", e);
```

```
    return 0;
}
```

Find the area of rectangle

```
#include<stdio.h>

int main(){
    int length=3, breadth=8;
    int area = length*breadth;
    printf("The area of this rectangle is %d", area);
    return 0;
}
```

Take the length and breath of rectangle from user through the keyboard and find the area of rectangle.

```
#include<stdio.h>

int main(){
    int length, breadth;
    printf("What is the length of the rectangle\n");
    scanf("%d", &length);

   printf("What is the breadth of the rectangle\n");
    scanf("%d", &breadth);

    printf("The area of your rectangle is %d", length*breadth);
    return 0;
}
```

Find the area of circle and volume of cylinder

```
#include <stdio.h>

int main()
{
```

```
    int radius = 3;
    float pi = 3.14;
    printf("The area of this circle is %f\n", pi * radius * radius);
    int height = 3;
    printf("Volume of this cylinder is %f\n", pi * radius * radius * height);
    return 0;
}
```

program to convert Celsius temperature in Fahrenheit

```
#include<stdio.h>

int main(){
    float celsius = 37, far;
    far = (celsius * 9 / 5) + 32;
    printf("The value of this celsius temperature in Fahrenheit is %f", far);
    return 0;
}
```

Program to calculate the simple interest for a given set of values years, rate of interest and principal.

```
#include<stdio.h>

int main(){
    int principal=100, rate=4, years=1;
    int simpleInterest = (principal * rate * years)/100;
    printf("The value of simple Interest is %d", simpleInterest);
    return 0;
}
```

Program to perform arithmetic instructions

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

int main(){
```

```
int a = 4;
int b = 8;

printf("The value of a + b is: %d\n", a + b);
printf("The value of a - b is: %d\n", a - b);
printf("The value of a * b is: %d\n", a * b);
printf("The value of a / b is: %d\n", a / b);

int z;
z = b * a; // legal
//b * a = z; // Illegal
printf("The value of z is: %d\n", z);

printf("5 when divided by 2 leaves a remainder of %d\n", 5%2);
printf("-5 when divided by 2 leaves a remainder of %d\n", -5%2);
printf("5 when divided by -2 leaves a remainder of %d\n", 5%-2);

// No operator is assumed to be present
//printf("The value of 4 * 5 is %d\n", (4)(5)); --> Will not return 20/ Wrong!!
printf("The value of 4 * 5 is %d\n", (4)*(5));

// There is no operator to perform exponentiation in C
printf("The value of 4 ^ 5 is %d\n", 4^5); // -> Will not produce 4 to the power 5
printf("The value of 4 to the power 5 is %f\n", pow(2, 5));


printf("The value of 6 + 5 is %d\n", 6 + 5);
printf("The value of 6 + 5.6 is %f\n", 6 + 5.6);
printf("The value of 6.1 + 5.6 is %f\n", 6.1 + 5.6);
printf("The value of 5/2 is %d\n", 5/2);
printf("The value of 3.0/9 is %f\n", 3.0 / 9);
```

```
    return 0;

}
```

Progran of Working of relational operators

```
#include <stdio.h>

int main()

{

    int a = 5, b = 5, c = 10;


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

    printf("%d == %d is %d \n", a, c, a == c);

    printf("%d > %d is %d \n", a, b, a > b);

    printf("%d > %d is %d \n", a, c, a > c);

    printf("%d < %d is %d \n", a, b, a < b);

    printf("%d < %d is %d \n", a, c, a < c);

    printf("%d != %d is %d \n", a, b, a != b);

    printf("%d != %d is %d \n", a, c, a != c);

    printf("%d >= %d is %d \n", a, b, a >= b);

    printf("%d >= %d is %d \n", a, c, a >= c);

    printf("%d <= %d is %d \n", a, b, a <= b);

    printf("%d <= %d is %d \n", a, c, a <= c);


    return 0;

}
```

Program of Working of logical operators

```
#include <stdio.h>

int main()

{

    int a = 5, b = 5, c = 10, result;
```

```
    result = (a == b) && (c > b);
    printf("(a == b) && (c > b) is %d \n", result);

    result = (a == b) && (c < b);
    printf("(a == b) && (c < b) is %d \n", result);

    result = (a == b) || (c < b);
    printf("(a == b) || (c < b) is %d \n", result);

    result = (a != b) || (c < b);
    printf("(a != b) || (c < b) is %d \n", result);

    result = !(a != b);
    printf("!(a != b) is %d \n", result);

    result = !(a == b);
    printf("!(a == b) is %d \n", result);

    return 0;
}
```

Program of Working of assignment operators

```
#include <stdio.h>
int main()
{
    int a = 5, c;

    c = a;     // c is 5
    printf("c = %d\n", c);
    c += a;    // c is 10
    printf("c = %d\n", c);
```

```
    c -= a;     // c is 5
    printf("c = %d\n", c);
    c *= a;     // c is 25
    printf("c = %d\n", c);
    c /= a;     // c is 5
    printf("c = %d\n", c);
    c %= a;     // c = 0
    printf("c = %d\n", c);

    return 0;
}
```

Program of Working of increment and decrement operators

```
#include <stdio.h>
int main()
{
    int a = 10, b = 100;
    float c = 10.5, d = 100.5;
    printf("++a = %d \n", ++a);
    printf("--b = %d \n", --b);
    printf("++c = %f \n", ++c);
    printf("--d = %f \n", --d);

    return 0;
}
```

Program of operator precedence.

```
#include <stdio.h>

int main()
{
```

```
    int x = 2;
    int y = 3;

    printf("The value of 3*x - 8*y is %d \n", 3*x - 8*y);
    printf("The value of 8*y / 3*x is %d \n", 8 * y / 3 * x);
    // 8*3 /3*x = 24/6 will give wrong answer
    // 24/3*2
    // 8*2
    // 16

    return 0;
}
```

Program of type declaration instructions.

```
#include<stdio.h>

int main(){
    int a = 4; // Type declaration instruction
     int a = 4, b, c; // Type declaration instruction
     b = c = a;

     printf("The value of a is %d\n", a);
     printf("The value of b is %d\n", b);
     printf("The value of c is %d\n", c);

    float a = 1.1;
    float b = a + 8.9;
    printf("The value of b is %f\n", b);
   return 0;
}
```

C Program to check whether a number is even or odd

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

```
int main(){
   int a, b;
   printf("Enter a number\n");
   scanf("%d", &a);

   if(a%2==0)
{
      printf("%d is even\n", a);
   }
   Else
{
      printf("%d is odd\n", a);
   }
   return 0;
}
```

Use of if_else statement

```
#include<stdio.h>

int main(){
   int age;
   printf("Enter your age\n");
   scanf("%d", &age);
   // if(age!=90){
   if(age>=90){
      printf("You are above 90, you cannot drive\n");
   }
   Else
{
      printf("You can drive\n");
   }
```

```
    if(age==50){
        printf("Half Century\n");
    }

    return 0;
}
```

Use of if_else_if statement

```
#include<stdio.h>

int main(){
    int num;

    printf("Enter your number\n");
    scanf("%d", &num);

    if(num==1){
        printf("Your number is 1\n");
    }
    else if (num == 2)
    {
        printf("Your number is 2\n");
    }
    else if (num == 3)
    {
        printf("Your number is 3\n");
    }
    else{
        printf("Its not 1, 2 or 3!\n");
    }
```

```
    return 0;
}
```

Use of logical operator

```
#include<stdio.h>

int main(){
    int age;
    int vipPass = 0;
    // vipPass = 1;

    printf("Enter your age\n");
    scanf("%d", &age);

    // if (age <= 70 && age>=18)
    if ((age <= 70 && age>=18) || (vipPass==1))
    {
        printf("You are above 18 and below 70, you can drive\n");
    }
    else
    {
        printf("You cannot drive\n");
    }

    return 0;
}
```

Program to find income tax as income entered by user.(using and operator)

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

```
int main(){
   float tax = 0, income;
   printf("Enter your income\n");
   scanf("%f", &income);

   if(income>=250000 && income<=500000){
      tax = tax + 0.05 * (income - 250000);
   }

   if (income >= 500000 && income <= 1000000)
   {
      tax = tax + 0.20 * (income - 500000);
   }

   if (income >= 1000000)
   {
      tax = tax + 0.30 * (income - 1000000);
   }

   printf("Your net income tax to be paid by 26th of this month is %f\n", tax);

   return 0;
}
```

Program to check whether given character is lowercase character or not.

```
#include<stdio.h>

int main(){
   // 97-122 = a-z ASCII Values
   char ch;
   printf("Enter the character\n");
```

```
    scanf("%c", &ch);
    if(ch<=122 && ch>=97){
        printf("It is lowercase");
    }
    else{
        printf("It is not lowercase");
    }
    return 0;
}
```

Program of switch case statement.

```
#include<stdio.h>

int main(){
    int rating;
    printf("Enter your rating (1-5)\n");
    scanf("%d", &rating);
    switch(rating){
        case 1:
            printf("Your rating is 1\n");
            break;
        case 2:
            printf("Your rating is 2\n");
            break;
        case 3:
            printf("Your rating is 3\n");
            break;
        case 4:
            printf("Your rating is 4\n");
            break;
        case 5:
            printf("Your rating is 5\n");
```

```
            break;
        default :
            printf("Invalid rating!\n");
            break;
    }

    return 0;
}
```

Program of ternary operator

```
#include<stdio.h>

int main(){
    int a,b,c,d;
b=10;
c=20;
    printf("Enter a\n");
    scanf("%d", &a);
int d=a<5?b:c;
    // One liner
    (a < 5) ? printf("A is less than 5") : printf("A is not less than 5");
     return 0;
}
```

Use of while loop

```
#include<stdio.h>

int main(){
    int a;
    scanf("%d", &a);
    while(a<10){
    //     a = 11;
    // while(a>10){ ---> These two lines will lead to an infinite loop
```

```
        printf("%d\n", a);
        a++;
    }

    return 0;
}
```

Use of increament operator

```
#include<stdio.h>

int main(){
    int i = 5;
    printf("The value after i++ is %d\n", ++i);
printf("The value after i++ is %d\n", i++);

    i++; // ---> Pehle print fir increment
    ++i; // ---> Pehle increment fir print kare
    printf("The value of i is %d\n", i);

    i+=10; //--> Increments i by 10
    printf("The value of i is %d\n", i);
    return 0;
}
```

Use of do while loop

```
#include<stdio.h>

int main(){
    int i = 220;

    do{
```

```
        printf("The value of i is %d\n", i);
        i++;
    }while(i <10);

    return 0;
}
```

Use of for loop

```
#include<stdio.h>

int main(){
    for(int a=0; a<10; a++){
        printf("The value of a is %d \n", a+1);
    }
    return 0;
}
```

Use of break statement

```
#include<stdio.h>

int main(){
    int i = 0;
    do{
        printf("The value of i is %d\n", i);
        if(i==4){
            break;
        }
        i++;
    }while(i<10);

    return 0;
}
```

Use of table using goto statement

```
int main()
{
 int num,i=1;
 printf("Enter the number whose table you want to print?");
 scanf("%d",&num);
 table:
 printf("%d x %d = %d\n",num,i,num*i);
 i++;
 if(i<=10)
 {
 goto table;
 }
}
```

Use of continue statement

```
#include<stdio.h>

int main(){
   int skip=5, i=0;
   while(i<10){
      i++;
      if(i!=skip){
         continue;
      }
      else{
         printf("%d\n", i);
      }
   }
   return 0;
}
```

Multiplication of table

```
#include<stdio.h>


int main(){
   printf("****Multiplication table of 10****\n\n");
   for(int i=10;i;i--){
      printf("10 X %d = %d\n", i, 10*i);
   }
   return 0;
}
```

Sum of first 10 numbers using loop

```
#include<stdio.h>
int main(){
   int i=1, sum=0, n=10;
   // for(i=1; i<=n; i++){
   //     sum +=i;
   // }
   while( i<=n){
      sum +=i;
      i++;
   }
   printf("The value of sum(1 to 10) is %d", sum);
   return 0;
}
```

Program to find factorial of given number using loop

```
#include<stdio.h>
int main(){
   // factorial(4) - 1 * 2 * 3 * 4
   // factorial(6) - 1 * 2 * 3 * 4 * 5 * 6
```

```
    int i=0, n=7, factorial=1;
    for(i=1;i<=n;i++){
        factorial *=i;
    }
    printf("The value of factorial %d is %d", n, factorial);
    return 0;
}
```

Program to find whether given number is prime or not

```
#include<stdio.h>
int main(){
    // Prime Numbers = A prime number (or a prime) is a natural number greater than 1 that is not a product of two smaller natural numbers.
    // Disclaimer: This is not the best method to solve this problem
    int n = 2, prime=1;
    for(int i=2;i<n;i++){
        if (n%i==0 ){
            prime = 0;
            break;
        }
    }
    if (prime==0){
        printf("This is not a prime number");
    }
    else{
        printf("This is a prime number");
    }
    return 0;
}
```

Program of basic function

```
#include<stdio.h>
void display(); // Function prototype
```

```
int main()
{
   int a;
   printf("Initializing display function\n");
   display(); // Function Call
   printf("Display function finished its work\n");
   return 0;
}

// Function definition
void display()
{
   printf("This is display\n");
}
```

Function inside the function

```
#include<stdio.h>
void goodMorning();
void goodAfternoon();
void goodNight();

int main()
{
   goodMorning();
   return 0;
}

void goodMorning()
{
   printf("Good Morning Harry\n");
   goodAfternoon();
}
```

```
void goodAfternoon()
{
   printf("Good Afternoon Harry\n");
   goodNight();
}

void goodNight()
{
   printf("Good Night Harry\n");
}
```

Program of sum function

```
#include<stdio.h>
// sum is a function which takes a and b as input and returns an integer as an output
int sum(int a, int b); // function prototype declaration

int main(){
   int c;
   c = sum(2, 15); // function call
   printf("The value of c is %d\n", c);
   return 0;
}

int sum (int a, int b)
{
   int c;
   c = a + b;
   return c;
}
```

Program of area of square using power function

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

```
#include<math.h>


int main(){
   int side;
   printf("Enter the value of side\n");
   scanf("%d", &side);
   printf("The value of area is %f", pow(side,2));


   return 0;
}
```

Program to find factorial using recursion

```
#include<stdio.h>
int factorial(int x);


int main(){
   int a = 5;
   printf("The value of factorial %d is %d", a, factorial(a));
   return 0;
}
int factorial(int x)
{
   printf("Calling factorial(%d)\n", x);
   if (x==1 || x==0){
      return 1;
   }
   else{
      return x * factorial(x-1);
   }
}
```

Program to find sum of natural numbers using recursion

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

```
int sum(int n);

int main() {
   int number, result;

   printf("Enter a positive integer: ");
   scanf("%d", &number);

   result = sum(number);

   printf("sum = %d", result);
   return 0;
}

int sum(int n) {
   if (n != 0)
      // sum() function calls itself
      return n + sum(n-1);
   else
      return n;
}
```

Average of three numbers using function

```
#include<stdio.h>
float average(int a, int b, int c);

int main(){
   int a, b, c;
   printf("Enter the value of a\n");
   scanf("%d", &a);
```

```
    printf("Enter the value of b\n");
    scanf("%d", &b);
    printf("Enter the value of c\n");
    scanf("%d", &c);
    printf("The value of average is %f", average(a, b, c));
    return 0;
}

float average(int a, int b, int c){
    float result;
    result = (float)(a + b + c)/3;
    return result;
}
```

Calculate the force when mass is given

```
#include<stdio.h>
float force(float mass);
int main(){
    float m;
    printf("Enter the value of mass in kgs\n");
    scanf("%f", &m);
    printf("The value of force in Newton is %.2f\n", force(m));
    return 0;
}

float force(float mass){
    float result = mass * 9.8;
    return result;
}
```

Use of increment operator

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

```
int main(){
   int a =3;
   printf("%d %d %d", a, ++a, a++);
   return 0;
}
```

Output- 5 5 3

Description:- firstly a++ will be evaluated, secondly ++a then a. so

a++ -> first assign the 3 to a then increment a that is a=4

++a-> secondly increment the value of a which is 4 and assign to a that is a=5

a-> third display the value of a that is 5

Program of pointer

```
#include<stdio.h>

int main(){
   int i = 34;
   int *j = &i; // j will now store the address of i
   printf("The value of i is %d\n", i);
   printf("The value of i is %d\n", *j);
   printf("The address of i is %u\n", &i);
   printf("The address of i is %u\n", j);
   printf("The address of j is %u\n", &j);
   printf("The value of j is %u\n", *(&j));
   return 0;
}
```

Call by value

```
#include<stdio.h>
int sum (int a, int b);
int main(){
   int x=4, y=7;
   printf("The value of x and y is %d and %d\n", x, y);
```

```
    printf("The value of 4+7 is %d\n", sum(x, y));
    printf("The value of x and y after function call is %d and %d\n", x, y);
    return 0;
}
int sum (int a, int b){
    int c;
    c = a + b;
    b =3434;
    a = 23432;
    return c;
}
```

Call by reference

```
#include<stdio.h>
void wrong_swap(int a, int b);
void swap(int *a, int *b);

int main(){
    int x=3, y=4;
    printf("The value of x and y before swap is %d and %d\n", x, y);
    //wrong_swap(x, y); // will not work due to call by value
    swap(&x, &y); // will work due to call by reference
    printf("The value of x and y after swap is %d and %d\n", x, y);

    return 0;
}

void wrong_swap(int a, int b){
    int temp;
    temp = a;
    a = b;
```

```
    b = temp;
}

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

Call by value and call by reference

```
#include<stdio.h>

void sumAndAvg(int a, int b, int *sum, float *avg){
    *sum = a +b;
    *avg = (float)(*sum)/2;
}

int main(){
    int i, j, sum;
    float avg;
    i = 3;
    j = 6;
    sumAndAvg(i, j, &sum, &avg);
    printf("The value of sum is %d \n", sum);
    printf("The value of avg is %f \n", avg);

    return 0;
}
```

Pointer to Pointer

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

```
int main(){
   int i = 345;
   int *ptr;
   int **ptr_ptr;

   ptr = &i;
   ptr_ptr= &ptr;

   printf("The value of i is %d", **ptr_ptr);
   return 0;
}
```

Program of basic array

```
#include<stdio.h>

int main(){
   // Naive way to create 4 ints
   // int marks1, marks2, marks3, marks4;
   // marks1 = 34;
   // marks2 = 45;
   // marks3 = 67;
   // marks4 = 87;
   int marks[4];
   marks[0] = 34;
   marks[1] = 45;
   marks[2] = 34;
   marks[3] = 67;


   return 0;
```

```
}
```

Program of array inputs

```
#include<stdio.h>


int main(){
   int marks[4]; // allocate space for 4 integers


   printf("Enter the value of marks for student 1: ");
   scanf("%d", &marks[0]);
   printf("Enter the value of marks for student 2: ");
   scanf("%d", &marks[1]);
   printf("Enter the value of marks for student 3: ");
   scanf("%d", &marks[2]);
   printf("Enter the value of marks for student 4: ");
   scanf("%d", &marks[3]);


   printf("You have entered %d %d %d and %d", marks[0],
         marks[1], marks[2], marks[3]);
   return 0;
}
```

Array input using loop

```
#include<stdio.h>


int main(){
   int marks[5];


   for(int i=0; i<5; i++)
   {
      printf("Enter the value of marks for student %d: ", i+1);
      scanf("%d", &marks[i]);
   }
```

```
    for(int i=0; i<5; i++)
    {
        printf("The value of marks for student %d is: %d \n", i+1, marks[i]);
    }

    return 0;
}
```

Array initialization

```
#include<stdio.h>

int main(){
    // int a[] = {34,232,23};
    // printf("The value of a[0] is %d\n", a[0]);
    // printf("The value of a[1] is %d\n", a[1]);
    // printf("The value of a[2] is %d\n", a[2]);

    float a[] = {3.4,2.32,2.3};
    printf("The value of a[0] is %f\n", a[0]);
    printf("The value of a[1] is %f\n", a[1]);
    printf("The value of a[2] is %f\n", a[2]);
    return 0;
}
```

Program of multidimensional array consist of 3 students having 5 subjects, taking marks of subject through keyboard.

```
#include<stdio.h>

int main(){
```

```
    int n_students = 3;
    int n_subjects = 5;

    int marks[3][5];
    for(int i=0; i<n_students; i++){
        for(int j=0; j<n_subjects; j++){
            printf("Enter the marks of student %d in subject %d\n", i+1, j+1);
            scanf("%d", &marks[i][j]);
        }
    }

    for(int i=0; i<n_students; i++){
        for(int j=0; j<n_subjects; j++){
            printf("The marks of student %d in subject %d is: %d\n", i+1, j+1, marks[i][j]);
        }
    }

    return 0;
}
```

Program to Add Two Matrices

```
#include <stdio.h>
int main() {
  int r, c, a[100][100], b[100][100], sum[100][100], i, j;
  printf("Enter the number of rows (between 1 and 100): ");
  scanf("%d", &r);
  printf("Enter the number of columns (between 1 and 100): ");
  scanf("%d", &c);

  printf("\nEnter elements of 1st matrix:\n");
  for (i = 0; i < r; ++i)
```

```
  for (j = 0; j < c; ++j) {
    printf("Enter element a%d%d: ", i + 1, j + 1);
    scanf("%d", &a[i][j]);
  }

 printf("Enter elements of 2nd matrix:\n");
 for (i = 0; i < r; ++i)
  for (j = 0; j < c; ++j) {
    printf("Enter element b%d%d: ", i + 1, j + 1);
    scanf("%d", &b[i][j]);
  }

 // adding two matrices
 for (i = 0; i < r; ++i)
  for (j = 0; j < c; ++j) {
    sum[i][j] = a[i][j] + b[i][j];
  }

 // printing the result
 printf("\nSum of two matrices: \n");
 for (i = 0; i < r; ++i)
  for (j = 0; j < c; ++j) {
    printf("%d   ", sum[i][j]);
    if (j == c - 1) {
      printf("\n\n");
    }
  }

 return 0;
}
```

Multiplication of two Matices

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

Pointer arithmetic

```
#include<stdio.h>

int main(){
   // int i = 34;
   // int *ptr = &i;
   // printf("The value of ptr is %u\n", ptr);
   // // ptr = ptr - 1;
   // ptr = ptr + 1;
   // // ptr++;
   // // ptr++;
   // printf("The value of ptr is %u\n", ptr);
```

```
    // char c = '3';
    // char *ptr = &c;
    // printf("The value of ptr is %u\n", ptr);
    // // ptr = ptr - 1;
    // ptr = ptr + 1;
    // ptr = ptr + 1;
    // // ptr++;
    // // ptr++;
    // printf("The value of ptr is %u\n", ptr);

    float f = 3.4;
    float *ptr = &f;
    printf("The value of ptr is %u\n", ptr);
    // ptr = ptr - 1;
    ptr = ptr + 1;
    // ptr = ptr + 1;
    // ptr++;
    // ptr++;
    printf("The value of ptr is %u\n", ptr);



    return 0;
}
```

Array with pointer

```
#include<stdio.h>

int main(){
    int marks[4];
    int *ptr;
    // ptr = &marks[0];
```

```
    ptr = marks;
    for(int i =0; i<4; i++){
        printf("Enter the value of marks for student %d: \n", i+1);
        scanf("%d", ptr);
        ptr ++;
    }

    for(int i =0; i<4; i++){
        printf("The value of marks for student %d is %d \n",i+1, marks[i]);
    }
    return 0;
}
```

Program of array to function

```
#include<stdio.h>

// void printArray(int *ptr, int n){
//     for(int i=0; i<n; i++){
//         printf("The value of element %d is %d\n", i+1, *(ptr+i));
//     }
// }

void printArray(int ptr[], int n){
    for(int i=0; i<n; i++){
        printf("The value of element %d is %d\n", i+1, ptr[i]);
    }
    ptr[2] = 5555; // This value will be changes in arr array of main as well
}

int main(){
    int arr[] = {1,2,3543,34,3,645,23};
    printArray(arr, 7);
```

```
    printf("%d", arr[2]);
    return 0;
}
```

Output:-

The value of element 1 is 1

The value of element 2 is 2

The value of element 3 is 3543

The value of element 4 is 34

The value of element 5 is 3

The value of element 6 is 645

The value of element 7 is 23

5555

Program of multidimensional array consist of 3 students having 5 subjects, taking marks of subject through keyboard.

```
#include<stdio.h>

int main(){
    int n_students = 3;
    int n_subjects = 5;

    int marks[3][5];
    for(int i=0; i<n_students; i++){
        for(int j=0; j<n_subjects; j++){
            printf("Enter the marks of student %d in subject %d\n", i+1, j+1);
            scanf("%d", &marks[i][j]);
        }
    }

    for(int i=0; i<n_students; i++){
```

```
        for(int j=0; j<n_subjects; j++){
            printf("The marks of student %d in subject %d is: %d\n", i+1, j+1, marks[i][j]);
        }
    }

    return 0;
}
```

Pointer storing the address of array and pointer increment

```
#include<stdio.h>

int main(){
    int arr[10];
    // int *ptr = &arr[0];
    int *ptr = arr;
    ptr = ptr+2;
    if(ptr==&arr[2]){
        printf("These point to the same location in memory\n");
    }
    else{
        printf("These do not point to the same location in memory\n");
    }

    return 0;
}
```

Table of 5 using array

```
#include<stdio.h>

int main(){
    int mul[10];
    for(int i=0; i<10; i++){
        mul[i] = 5*(i+1);
```

```
    }

    for(int i=0; i<10; i++){
        printf("5X%d = %d\n", i+1, mul[i]);
    }
    return 0;
}
```

Program to reverse the array element using pointer

```
#include<stdio.h>
void reverse(int *arr, int n){
    int temp;
    for(int i=0; i<(n/2); i++){
        temp = arr[i];
        arr[i] = arr[n-i-1];
        arr[n-i-1] = temp;
    }
}

int main(){
    int arr[] = {1,2,3,4,5,6,7};
    reverse(arr, 7);
    for(int i=0; i<7; i++){
        printf("The value of %d element is: %d\n", i, arr[i]);
    }
    return 0;
}
```

Program of table using multidimensional array

```
#include<stdio.h>
void printTable(int *mulTable, int num, int n){
    printf("The multiplication table of %d is :\n", num);
    for(int i=0; i<n; i++){
```

```
        mulTable[i] = num*(i+1);
    }

    for(int i=0; i<n; i++){
        printf("%dX%d = %d\n", num, i+1, mulTable[i]);
    }
    printf("**********************************************************\n\n");
}

int main(){
    int mulTable[3][10];
    printTable(mulTable[0], 2, 10);
    printTable(mulTable[1], 7, 10);
    printTable(mulTable[2], 9, 10);

    return 0;
}
```

Program of three dimensional array

```
#include<stdio.h>

int main(){
    int arr[2][3][4];
    for(int i=0;i<2;i++){
        for(int j=0;j<3;j++){
            for(int k=0;k<4;k++){
                printf("The address of arr[%d][%d][%d] is %u\n", i, j, k, &arr[i][j][k]);
            }
        }
    }

    return 0;
```

```
}
```

Program introduction to string

```
#include<stdio.h>

int main(){

   // char str[] = {'H', 'a', 'r', 'r', 'y', '\0'};

   char str[] = "Harry";


   return 0;

}
```

Programme to display string using pointer

```
#include<stdio.h>


int main(){

   // char str[] = "Harry";

   char str[] = {'H', 'a', 'r', 'r', 'y', '\0'};

   char *ptr = str;

   while(*ptr!='\0'){

      printf("%c", *ptr);

      ptr++;

   }

   return 0;

}
```

Display string using array

```
#include<stdio.h>


int main(){

   // int a = 4;

   // printf("%d", a);
```

```
    // char *ptr = "B singh";
    char ptr[] = "B singh";
    printf("%s", ptr);
    return 0;
}
```

Input the string through keyboard

```
#include<stdio.h>

int main(){
    char s[34];
    printf("Enter your name: ");
    scanf("%s", s);
    printf("Your name is %s", s);
    return 0;
}
```

gets() and puts() function in string

```
#include<stdio.h>

int main(){
    char s[34];
    printf("Enter your name: ");
    gets(s);
    puts(s);
    // printf("Your name is %s", s);
    return 0;
}
```

Strlen() function

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

int main()
{
   char *st = "Harry";
   int a = strlen(st);
   printf("The length of string st is %d", a);
   return 0;
}
```

Strcpy() function

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

int main(){
   char *st = "This";
   char st2[45];
   strcpy(st2, st);
   printf("Now the st2 is %s", st2);
   return 0;
}
```

Strcmp() function

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

int main(){
   char st1[45] = "Hel";
   char *st2 = "Helo";
```

```
    int val = strcmp(st1, st2);
    printf("Now the val is %d", val);
    return 0;
}
```

Strcat() function

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

int main(){
    char st1[45] = "Hello";
    char *st2 = "Harry";
    strcat(st1, st2);
    printf("Now the st1 is %s", st1);
    return 0;
}
```

Write a function to find the length of string.

```
#include<stdio.h>
int strlen(char * st)
{
    char *ptr = st;
    int len=0;
    while(*ptr!='\0')
       {
       len++;
       ptr++;
    }
    return len;
}
```

```
int main()
{
   char st[] = "b Singh";
   int l = strlen(st);
   printf("The length of this string is %d", l);
   return 0;
}
```

Write a function to encrypt the given string by adding 1 to ASCII value of its characters.

```
#include<stdio.h>
void encrypt(char *c){
   char *ptr = c;
   while(*ptr!='\0'){
      *ptr = *ptr + 1;
      ptr++;
   }
}

int main(){
   char c[] = " Learn the AI Tools";
   encrypt(c);
   printf("Encrypted string is: %s", c);
   return 0;
}
```

Write a function to decrypt the given string in previous program.

```
#include<stdio.h>
void decrypt(char *c){
```

```
    char *ptr = c;
    while(*ptr!='\0'){
        *ptr = *ptr - 1;
        ptr++;
    }
}

int main(){
    char c[] = " Mfbso!uif!BJ!Uppmt";
    decrypt(c);
    printf("Decrypted string is: %s", c);
    return 0;
}
```

Write a program to count the occurrence of given character in string.

```
#include<stdio.h>
int occurence(char st[], char c){
    char *ptr = st;
    int count=0;
    while(*ptr!='\0'){
        if (*ptr==c){
            count++;
        }
        ptr++;
    }
    return count;
}
int main(){
    char st[] = "Welcome to the computer science technology";
    char c;
    printf("enter the character which you want to count\n");
    scanf("%c",&c);
```

```
    int count = occurence(st, c);
    printf("Occurences = %d", count);
    return 0;
}
```

Demonstration of structure in c programming.

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

struct employee{
    int code;
    float salary;
    char name[10];
};

int main(){
    int a =34;
    char b = 'g';
    float d = 234.3543;
    // employee e1;
    // e1.salary = 34.454; --->wont work without employee structure

    struct employee e1;
    e1.code = 100;
    e1.salary = 34.454;
    // e1.name = "Harry"; --> wont work
    strcpy(e1.name, "Harry");

    printf("%d\n", e1.code);
    printf("%.3f\n", e1.salary);
```

```
    printf("%s\n", e1.name);


    return 0;
}
```

Write a program of structure and enter the values through keyboard.

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

struct employee{
    int code;
    float salary;
    char name[10];
};

int main(){
    struct employee e1, e2, e3;
    printf("Enter the value for code of e1: ");
    scanf("%d", &e1.code);
    printf("Enter the value for salary of e1: ");
    scanf("%f", &e1.salary);
    printf("Enter the value for name of e1: ");
    scanf("%s", e1.name);

    printf("Enter the value for code of e2: ");
    scanf("%d", &e2.code);
    printf("Enter the value for salary of e2: ");
    scanf("%f", &e2.salary);
    printf("Enter the value for name of e2: ");
    scanf("%s", e2.name);
```

```
    printf("Enter the value for code of e3: ");
    scanf("%d", &e3.code);
    printf("Enter the value for salary of e3: ");
    scanf("%f", &e3.salary);
    printf("Enter the value for name of e3: ");
    scanf("%s", e3.name);

    return 0;
}
```

Array of structure

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

struct employee{
    int code;
    float salary;
    char name[20];
};

int main(){
    struct employee facebook[100];

    facebook[0].code = 100;
    facebook[0].salary = 100.45;
    strcpy(facebook[0].name, "Harry");

    facebook[1].code = 101;
    facebook[1].salary = 90.45;
    strcpy(facebook[1].name, "Rohan");
```

```
    facebook[2].code = 102;
    facebook[2].salary = 110.45;
    strcpy(facebook[2].name, "SkillKhiladi");
    printf("Done");

    return 0;
}
```

Initialization of structure variable.

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

struct employee{
    int code;
    float salary;
    char name[20];
};
int main(){
    struct employee bsingh = {100, 34.23, "bsingh"};
    printf("Code is: %d \n", bsingh.code);
    printf("Salary is: %f \n", bsingh.salary);
    printf("Name is: %s \n", bsingh.name);
     return 0;
}
```

Pointer to the structure

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

struct employee{
```

```
    int code;
    float salary;
    char name[20];
};

int main(){
    struct employee e1;
    struct employee *ptr;

    ptr = &e1;
    //(*ptr).code = 101; //or you can also write: ptr->code = 101;
    ptr->code = 101;
    printf("%d", e1.code);

    return 0;
}
```

Structure with function to display the values of structure variables.

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

struct employee{
    int code;
    float salary;
    char name[20];
};

void show(struct employee emp){
    printf("The Code of employee is: %d\n", emp.code);
    printf("The Salary of employee is: %f\n", emp.salary);
    printf("The Name of employee is: %s\n", emp.name);
```

```
    emp.code = 34;
}
int main(){
    struct employee e1;
    struct employee *ptr;

    ptr = &e1;
    //(*ptr).code = 101; //or you can also write: ptr->code = 101;
    ptr->code = 101;
    ptr->salary = 11.01;
    strcpy(ptr->name, "Harry");

    show(e1);
    printf("The Code of employee is: %d\n", e1.code);

    return 0;
}
```

Typedef in structure

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

typedef struct employee{
    int code;
    float salary;
    char name[20];
} emp;

void show(emp emp1){
    printf("The Code of employee is: %d\n", emp1.code);
```

```
    printf("The Salary of employee is: %f\n", emp1.salary);
    printf("The Name of employee is: %s\n", emp1.name);

}

int main(){
    // Declaring e1 and ptr
    emp e1;
    emp *ptr;

    // pointing ptr to e1
    ptr = &e1;

    // Set the member values for e1
    ptr->code = 101;
    ptr->salary = 11.01;
    strcpy(ptr->name, "Harry");

    show(e1);

    return 0;
}
```

Create a two dimensional vector using structure

```
#include<stdio.h>

struct vector{
    int x;
    int y;
};
```

```
int main(){
    struct vector v1, v2;
    v1.x = 34;
    v1.y = 54;
    printf("X dim is %d and Y dim is %d\n", v1.x, v1.y);

    v2.x = 3345;
    v2.y = 534;
    printf("X dim is %d and Y dim is %d\n", v2.x, v2.y);
    return 0;
}
```

Write a function to find sum of two dimensional vector.

```
#include<stdio.h>

struct vector{
    int x;
    int y;
};

struct vector sumVector(struct vector v1, struct vector v2){
    struct vector result;
    result.x = v1.x + v2.x;
    result.y = v1.y + v2.y;
    return result;
}
int main(){
    struct vector v1, v2, sum;
    v1.x = 4;
    v1.y = 9;
    printf("X dim is %d and Y dim is %d\n", v1.x, v1.y);
```

```
    v2.x = 5;
    v2.y = 4;
    printf("X dim is %d and Y dim is %d\n", v2.x, v2.y);

    sum = sumVector(v1, v2);
    printf("X dim of result is %d and Y dim is %d\n", sum.x, sum.y);
    return 0;
}
```

Create a structure with complex numbers and display with display function. The values must be taken from the user.

```
#include<stdio.h>

typedef struct complex{
    int real;
    int complex;
}comp;

void display(comp c){
    printf("The value of real part is %d\n", c.real);
    printf("The value of Imaginary part is %d\n", c.complex);
}

int main(){
    comp cnums[5];
    for(int i=0; i<5; i++){
        printf("Enter the real value for %d num\n", i+1);
        scanf("%d", &cnums[i].real);

        printf("Enter the complex value for %d num\n", i+1);
```

```
        scanf("%d", &cnums[i].complex);
    }
    for(int i=0; i<5; i++){
        display(cnums[i]);
    }
    return 0;
}
```