1. C program to perform all arithmetic operation

#include <stdio.h>

int main()

{

int num1, num2;

int sum, sub, mult, mod;

float div;

printf("Enter any two numbers: ");

scanf("%d%d", &num1, &num2);

sum = num1 + num2;

sub = num1 - num2;

mult = num1 \* num2;

div = (float)num1 / num2;

mod = num1 % num2;

printf("SUM = %d\n", sum);

printf("DIFFERENCE = %d\n", sub);

printf("PRODUCT = %d\n", mult);

printf("QUOTIENT = %f\n", div);

printf("MODULUS = %d", mod);

return 0;

}

1. C program to find area of a triangle if base and height are given.

#include <stdio.h>

int main()

{

float base, height, area;

printf("Enter base of the triangle: ");

scanf("%f", &base);

printf("Enter height of the triangle: ");

scanf("%f", &height);

area = (base \* height) / 2;

printf("Area of the triangle = %.2f sq. units", area);

return 0;

}

1. C program to find all angles of a triangle if two angles are given

#include <stdio.h>

int main()

{

int a, b, c;

printf("Enter two angles of triangle: ");

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

c = 180 - (a + b);

printf("Third angle of the triangle = %d", c);

return 0;

}

1. C program to convert days in to years, weeks and days.

#include <stdio.h>

int main()

{

int days, years, weeks;

printf("Enter days: ");

scanf("%d", &days);

years = (days / 365);

weeks = (days % 365) / 7;

days = days - ((years \* 365) + (weeks \* 7));

printf("YEARS: %d\n", years);

printf("WEEKS: %d\n", weeks);

printf("DAYS: %d", days);

return 0;

}

1. C program to find power and square root of any number.

#include <stdio.h>

#include <math.h>

int main()

{

double num, root;

printf("Enter any number to find square root: ");

scanf("%lf", &num);

root = sqrt(num);

printf("Square root of %.2lf = %.2lf", num, root);

return 0;

}

1. C program to calculate total, average and percentage and grades of five subjects.

#include <stdio.h>

int main()

{

float eng, phy, chem, math, comp;

float total, average, percentage;

printf("Enter marks of five subjects: \n");

scanf("%f%f%f%f%f", &eng, &phy, &chem, &math, &comp);

total = eng + phy + chem + math + comp;

average = total / 5.0;

percentage = (total / 500.0) \* 100

printf("Total marks = %.2f\n", total);

printf("Average marks = %.2f\n", average);

printf("Percentage = %.2f", percentage);

return 0;

}

1. C program to check Least Significant Bit (LSB) of a number using bitwise operator.

#include <stdio.h>

int main()

{

int num;

printf("Enter any number: ");

scanf("%d", &num);

if(num & 1)

printf("LSB of %d is set (1).", num);

else

printf("LSB of %d is unset (0).", num);

return 0;

}

1. C program to check most significant bit ( MSB) of a number using bitwise operator.

#include <stdio.h>

#define BITS sizeof(int)

int main()

{

int num, msb;

printf("Enter any number: ");

scanf("%d", &num);

msb = 1 << (BITS - 1);

if(num & msb)

printf("MSB of %d is set (1).", num);

else

printf("MSB of %d is unset (0).", num);

return 0;

}

1. C program to swap two numbers USING 3RD VARIABLE AND WITHOUT 3RD VARIABLE.

Using

#include<stdio.h>

int main() {

int x, y, temp;

printf("Enter the value of x and y: ");

scanf("%d %d", &x, &y);

printf("Before swapping x=%d, y=%d ", x, y);

temp =x;

x = y;

y = temp;

printf("After swapping x=%d, b=%d", x, y);

return 0;

}

Without using third variable

#include<stdio.h>

int main()

{

int a, b;

printf("Enter value of A: ");

scanf("%d", & a);

printf("Enter value of B: ");

scanf("%d", & b);

printf("A = %d, B = %d", a, b);

a = a + b;

b = a - b;

a = a - b;

printf("\nNow, A = %d, B = %d", a, b);

}

1. C program to find maximum between three numbers using

conditional operator

#include <stdio.h>

int main()

{

int num1, num2, num3, max;

printf("Enter three numbers: ");

scanf("%d%d%d", &num1, &num2, &num3);

max = (num1 > num2 && num1 > num3) ? num1:

(num2 > num3) ? num2 : num3;

printf("\nMaximum between %d, %d and %d = %d", num1, num2, num3, max);

return 0;

}

1. C program to check alphabet, digit or special character using conditional operator

#include<stdio.h>

int main()

{

char ch;

printf("enter character :");

scanf("%c",&ch);

if(ch>=65 && ch<=90 || ch>=97 && ch<=122)

  {

printf("character is alphabet");

     }

   else if(ch>=48 && ch<=57)

printf("character is number");

    else

printf("special character");

}

1. C program to calculate total electricity bill

#include <stdio.h>

int main()

{

int unit;

float amt, total\_amt, sur\_charge;

printf("Enter total units consumed: ");

scanf("%d", &unit);

if(unit <= 50)

{

amt = unit \* 0.50;

}

else if(unit <= 150)

{

amt = 25 + ((unit-50) \* 0.75);

}

else if(unit <= 250)

{

amt = 100 + ((unit-150) \* 1.20);

}

else

{

amt = 220 + ((unit-250) \* 1.50);

}

sur\_charge = amt \* 0.20;

total\_amt = amt + sur\_charge;

printf("Electricity Bill = Rs. %.2f", total\_amt);

return 0;

}

1. C program to create Simple Calculator AND Days of week

using switch case

#include <stdio.h>

int main()

{

int week;

printf("Enter week number(1-7): ");

scanf("%d", &week);

switch(week)

{

case 1:

printf("Monday");

break;

case 2:

printf("Tuesday");

break;

case 3:

printf("Wednesday");

break;

case 4:

printf("Thursday");

break;

case 5:

printf("Friday");

break;

case 6:

printf("Saturday");

break;

case 7:

printf("Sunday");

break;

default:

printf("Invalid input! Please enter week number between 1-7.");

}

return 0;

}

1. C program to check vowel or consonant using switch case.

#include <stdio.h>

int main()

{

char ch;

printf("Enter any alphabet: ");

scanf("%c", &ch);

switch(ch)

{

case 'a':

printf("Vowel");

break;

case 'e':

printf("Vowel");

break;

case 'i':

printf("Vowel");

break;

case 'o':

printf("Vowel");

break;

case 'u':

printf("Vowel");

break;

case 'A':

printf("Vowel");

break;

case 'E':

printf("Vowel");

break;

case 'I':

printf("Vowel");

break;

case 'O':

printf("Vowel");

break;

case 'U':

printf("Vowel");

break;

default:

printf("Consonant");

}

return 0;

}

1. C program to check positive negative or zero using switch case

#include <stdio.h>

int main()

{

int num;

printf("Enter any number: ");

scanf("%d", &num);

switch (num > 0)

{

case 1:

printf("%d is positive.", num);

break;

case 0:

switch (num < 0)

{

case 1:

printf("%d is negative.", num);

break;

case 0:

printf("%d is zero.", num);

break;

}

break;

}

return 0;

}

1. C program to check whether a triangle is Equilateral, Isosceles or Scalene.

#include <stdio.h>

int main()

{

int side1, side2, side3;

printf("Enter three sides of triangle: ");

scanf("%d%d%d", &side1, &side2, &side3);

if(side1==side2 && side2==side3)

{

printf("Equilateral triangle.");

}

else if(side1==side2 || side1==side3 || side2==side3)

{

printf("Isosceles triangle.");

}

else

{

printf("Scalene triangle.");

}

return 0;

}

1. C program to print all natural numbers AND sum of it from 1 To n

#include <stdio.h>

int main() {

int n, i, sum = 0;

printf("Enter a positive integer: ");

scanf("%d", &n);

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

sum += i;

}

printf("Sum = %d", sum);

return 0;

}

1. C program to print all even numbers AND sum of it from 1 to n.

#include <stdio.h>

int main()

{

int i, n, sum=0;

printf("Enter upper limit: ");

scanf("%d", &n);

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

{

sum += i;

}

printf("Sum of all even number between 1 to %d = %d", n, sum);

return 0;

}

1. C program to print multiplication table of a number.

#include <stdio.h>

int main()

{

int i, num;

printf("Enter number to print table: ");

scanf("%d", &num);

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

{

printf("%d \* %d = %d\n", num, i, (num\*i));

}

return 0;

}

1. C program to calculate factorial of a number.

#include<stdio.h>

int main()

{

 int i,fact=1,number;

 printf("Enter a number: ");

  scanf("%d",&number);

    for(i=1;i<=number;i++){

      fact=fact\*i;

  }

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

return 0;

}

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

#include <stdio.h>

int main()

{

int n, num, rev = 0;

printf("Enter any number to check palindrome: ");

scanf("%d", &n);

num = n;

while(n != 0)

{

rev = (rev \* 10) + (n % 10);

n /= 10;

}

if(rev == num)

{

printf("%d is palindrome.", num);

}

else

{

printf("%d is not palindrome.", num);

}

return 0;

}

1. C program to count frequency of digits in a given number.

#include <stdio.h>

#define BASE 10

int main()

{

long long num, n;

int i, lastDigit;

int freq[BASE];

printf("Enter any number: ");

scanf("%lld", &num);

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

{

freq[i] = 0;

}

n = num;

while(n != 0)

{

lastDigit = n % 10;

n /= 10;

freq[lastDigit]++;

}

printf("Frequency of each digit in %lld is: \n", num);

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

{

printf("Frequency of %d = %d\n", i, freq[i]);

}

return 0;

}

1. C program to print Armstrong numbers from 1 to n AND

Check a given number is Armstrong numbers or not.

#include<stdio.h>

int main()

{

int n,r,sum=0,temp;

printf("enter the number=");

scanf("%d",&n);

temp=n;

while(n>0)

{

r=n%10;

sum=sum+(r\*r\*r);

n=n/10;

}

if(temp==sum)

printf("armstrong number ");

else

printf("not armstrong number");

return 0;

}

1. C program to find HCF(GCD) AND LCM of two numbers.

#include <stdio.h>

int main() {

int a, b, x, y, t, hcf, lcm;

printf("Enter two integers\n");

scanf("%d%d", &x, &y);

a = x;

b = y;

while (b != 0) {

t = b;

b = a % b;

a = t;

}

hcf = a;

lcm = (x\*y)/hcf;

printf("highest common factor of %d and %d = %d\n", x, y, hcf);

printf("Least common multiple of %d and %d = %d\n", x, y, lcm);

return 0;

}

1. C program to print all prime numbers between 1 to n.

#include <stdio.h>

int main()

{

int i, j, end, isPrime;

printf("Find prime numbers between 1 to : ");

scanf("%d", &end);

printf("All prime numbers between 1 to %d are:\n", end);

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

{

isPrime = 1;

for(j=2; j<=i/2; j++)

{

if(i%j==0)

{

isPrime = 0;

break;

}

}

if(isPrime==1)

{

printf("%d, ", i);

}

}

return 0;

}

1. C program to print all Strong Numbers between 1 to n

#include <stdio.h>

int main()

{

int i, j, cur, lastDigit, end;

long long fact, sum;

printf("Enter upper limit: ");

scanf("%d", &end);

printf("All Strong numbers between 1 to %d are:\n", end);

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

{

cur = i;

sum = 0;

while(cur > 0)

{

fact = 1;

lastDigit = cur % 10;

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

{

fact = fact \* j;

}

sum += fact;

cur /= 10;

}

if(sum == i)

{

printf("%d, ", i);

}

}

return 0;

}

1. C program to print Fibonacci series up to n terms.

#include <stdio.h>

int main()

{

int a, b, c, i, terms;

printf("Enter number of terms: ");

scanf("%d", &terms);

a = 0;

b = 1;

c = 0;

printf("Fibonacci terms: \n");

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

{

printf("%d, ", c);

a = b;

b = c;

c = a + b;

}

return 0;

}

1. C program to print all Perfect numbers between 1 to n AND

Check a given number is Perfect numbers or not.

#include <stdio.h>

int main()

{

int i, j, end, sum;

printf("Enter upper limit: ");

scanf("%d", &end);

printf("All Perfect numbers between 1 to %d:\n", end);

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

{

sum = 0;

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

{

if(i % j == 0)

{

sum += j;

}

}

if(sum == i)

{

printf("%d, ", i);

}

}

return 0;

}

1. C program to find power of any number using for loop.

#include <stdio.h>

int main()

{

int base, exponent;

long long power = 1;

int i;

printf("Enter base: ");

scanf("%d", &base);

printf("Enter exponent: ");

scanf("%d", &exponent);

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

{

power = power \* base;

}

printf("%d ^ %d = %lld", base,exponent,power);

return 0;

}

1. C program to print ASCII values of all characters.

#include <stdio.h>

int main()

{

int i;

for(i=0; i<=255; i++)

{

printf("ASCII value of character %c = %d\n", i, i);

}

return 0;

}

1. C program to print Pascal triangle up to n rows.

#include <stdio.h>

long long fact(int n);

int main()

{

int n, k, num, i;

long long term;

printf("Enter number of rows : ");

scanf("%d", &num);

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

{

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

printf("%3c", ' ');

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

{

term = fact(n) / (fact(k) \* fact(n-k));

printf("%6lld", term);

}

printf("\n");

}

return 0;

}

long long fact(int n)

{

long long factorial = 1ll;

while(n>=1)

{

factorial \*= n;

n--;

}

return factorial;

}

1. C program to find sum of all elements of array.

#include<stdio.h>

int main()

{

int arr[100], size, i, sum = 0;

printf("Enter array size\n");

scanf("%d",&size);

printf("Enter array elements\n");

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

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

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

sum = sum + arr[i];

printf("Sum of the array = %d\n",sum);

return 0;

}

1. C program to copy one array to another array.

include <stdio.h>

#define MAX\_SIZE 100

int main()

{

int source[MAX\_SIZE], dest[MAX\_SIZE];

int i, size;

printf("Enter the size of the array : ");

scanf("%d", &size);

printf("Enter elements of source array : ");

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

{

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

}

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

{

dest[i] = source[i];

}

printf("\nElements of source array are : ");

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

{

printf("%d\t", source[i]);

}

printf("\nElements of dest array are : ");

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

{

printf("%d\t", dest[i]);

}

return 0;

}

Bottom of Form

1. C program to insert an element in array at specified position.

include <stdio.h>

#define MAX\_SIZE 100

int main()

{

int arr[MAX\_SIZE];

int i, size, num, pos;

printf("Enter size of the array : ");

scanf("%d", &size);

printf("Enter elements in array : ");

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

{

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

}

printf("Enter element to insert : ");

scanf("%d", &num);

printf("Enter the element position : ");

scanf("%d", &pos);

if(pos > size+1 || pos <= 0)

{

printf("Invalid position! Please enter position between 1 to %d", size);

}

else

{

for(i=size; i>=pos; i--)

{

arr[i] = arr[i-1];

}

arr[pos-1] = num;

size++;

printf("Array elements after insertion : ");

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

{

printf("%d\t", arr[i]);

}

}

return 0;

}

1. C program to delete an element in array at specified position.

include <stdio.h>

#define MAX\_SIZE 100

int main()

{

int arr[MAX\_SIZE];

int i, size, pos;

printf("Enter size of the array : ");

scanf("%d", &size);

printf("Enter elements in array : ");

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

{

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

}

printf("Enter the element position to delete : ");

scanf("%d", &pos);

if(pos < 0 || pos > size)

{

printf("Invalid position! Please enter position between 1 to %d", size);

}

else

{

for(i=pos-1; i<size-1; i++)

{

arr[i] = arr[i + 1];

}

size--;

printf("\nElements of array after delete are : ");

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

{

printf("%d\t", arr[i]);

}

}

return 0;

}

1. C program to search element in array using Linear Search.

include <stdio.h>

int main()

{

int array[100], search, c, num;

printf("Enter the number of elements in array\n");

scanf("%d",&num);

printf("Enter %d numbers\n", num);

for ( c = 0 ; c < number ; c++ )

scanf("%d",&array[c]);

printf("Enter the number to search\n");

scanf("%d",&search);

for ( c = 0 ; c <num ; c++ )

{

if ( array[c] == search )

{

printf("%d is present at location %d.\n", search, c+1);

break;

}

}

if ( c == num )

printf("%d is not present in array.\n", search);

return 0;

}

1. C program to find second largest number and Sorting Using Bubble sort in an array

include<stdio.h>

int main()

{

int a[10],i,j,temp,n;

printf("\n Enter the max no.of Elements to Sort: \n");

scanf("%d",&n);

printf("\n Enter the Elements : \n");

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

{

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

}

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

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

{

if(a[i]>a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

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

{

printf("%d\t",a[i]);

}

return 0;

}

1. C program to count total number of duplicate elements in an Array

include <stdio.h>

#define MAX\_SIZE 100

int main()

{

int arr[MAX\_SIZE];

int i, j, size, count = 0;

printf("Enter size of the array : ");

scanf("%d", &size);

printf("Enter elements in array : ");

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

{

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

}

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

{

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

{

if(arr[i] == arr[j])

{

count++;

break;

}

}

}

printf("\nTotal number of duplicate elements found in array = %d", count);

return 0;

}

1. C program to perform scalar matrix multiplication.

include <stdio.h>

#define SIZE 3

int main()

{

int A[SIZE][SIZE];

int num, row, col;

printf("Enter elements in matrix of size %dx%d: \n", SIZE, SIZE);

for(row=0; row<SIZE; row++)

{

for(col=0; col<SIZE; col++)

{

scanf("%d", &A[row][col]);

}

}

printf("Enter any number to multiply with matrix A: ");

scanf("%d", &num);

for(row=0; row<SIZE; row++)

{

for(col=0; col<SIZE; col++)

{

A[row][col] = num \* A[row][col];

}

}

printf("\nResultant matrix c.A = \n");

for(row=0; row<SIZE; row++)

{

for(col=0; col<SIZE; col++)

{

printf("%d ", A[row][col]);

}

printf("\n");

}

return 0;

}

1. C program to find sum of main diagonal elements of a matrix.

include <stdio.h>

#define SIZE 3

int main()

{

int A[SIZE][SIZE];

int row, col, sum = 0;

printf("Enter elements in matrix of size %dx%d: \n", SIZE, SIZE);

for(row=0; row<SIZE; row++)

{

for(col=0; col<SIZE; col++)

{

scanf("%d", &A[row][col]);

}

}

for(row=0; row<SIZE; row++)

{

sum = sum + A[row][row];

}

printf("\nSum of main diagonal elements = %d", sum);

return 0;

}

1. C program to check whether a matrix is Identity matrix or not.

#include <stdio.h>

int main (void)

{

int a[10][10];

int i = 0, j = 0, row = 0, col = 0;

printf ("Enter the order of the matrix (mxn):\n");

printf ("where m = number of rows; and\n");

printf (" n = number of columns\n");

scanf ("%d %d", &row, &col);

int flag = 0;

printf ("Enter the elements of the matrix\n");

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++)

{

if (i == j && a[i][j] != 1)

{

flag = -1;

break;

}

else if (i != j && a[i][j] != 0)

{

flag = -1;

break;

}

}

}

if (flag == 0)

{

printf ("It is a IDENTITY MATRIX\n");

}

else

{

printf ("It is NOT an identity matrix\n");

}

return 0;

}

1. C program to merge two sorted array in ascending order.[doubtful]

#include<stdio.h>

int main( )

{

int a[50], b[25], i, j, k=1, s, m, n, temp ;

printf(" Enter the number of element in first array : ") ;

scanf("%d ",&m) ;

printf("\n Enter the element of first array in ascending order : \n") ;

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

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

printf(" Enter the number of element in second array : ") ;

scanf("%d ",&n) ;

printf("\n Enter the element of second array in ascending order : \n") ;

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

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

s = m + n ;

for ( i = m+1 ; i<=s ; i++)

{

a[i] = b[k] ;

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

{

if ( a[j] >= a[i] )

{

temp = a[i] ;

a[j] = a[i] ;

a[i] = temp ;

}

}

k = k+1 ;

}

printf("\n Array after merging :\n") ;

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

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

return ( 0 ) ;

}

1. C program for all operations in string to check palindrome number or not.

#include <stdio.h>

#include <string.h>

void main()

{

char string[25], reverse\_string[25] = {'0'};

int i, length = 0, flag = 0;

fflush(stdin);

printf("Enter a string: \n");

gets(string);

for (i = 0; string[i] != '\0'; i++)

{

length++;

}

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

{

reverse\_string[length - i - 1] = string[i];

}

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

{

if (reverse\_string[i] == string[i])

flag = 1;

else

flag = 0;

}

if (flag == 1)

printf("%s is a palindrome \n", string);

else

printf("%s is not a palindrome \n", string);

}

1. C program to check whether a string is palindrome or not .without Compare Function of String.

#include<stdio.h>

int main()

{

char string[40];

int length=0, flag=1,i;

printf("Enter string:\n");

gets(string);

for(i=0;string[i]!='\0';i++)

{

length++;

}

for(i=0;i< length/2;i++)

{

if( string[i] != string[length-1-i] )

{

flag=0;

break;

}

}

if(flag==1)

{

printf("PALINDROME");

}

else

{

printf("NOT PALINDROME");

}

return 0;

}

1. C program to count frequency of each character in a string.

#include <stdio.h>

int main() {

char str[1000], ch;

int count = 0;

printf("Enter a string: ");

gets(str, sizeof(str), stdin);

printf("Enter a character to find its frequency: ");

scanf("%c", &ch);

for (int i = 0; str[i] != '\0'; ++i) {

if (ch == str[i])

++count;

}

printf("Frequency of %c = %d", ch, count);

return 0;

}

1. C program to find diameter, circumference and area of a circle

using functions.

#include <stdio.h>

#include <math.h>

double getDiameter(double radius);

double getCircumference(double radius);

double getArea(double radius);

int main()

{

float radius, dia, circ, area;

printf("Enter radius of circle: ");

scanf("%f", &radius);

dia = getDiameter(radius);

circ = getCircumference(radius);

area = getArea(radius);

printf("Diameter of the circle = %.2f units\n", dia);

printf("Circumference of the circle = %.2f units\n", circ);

printf("Area of the circle = %.2f sq. units", area);

return 0;

}

double getDiameter(double radius)

{

return (2 \* radius);

}

double getCircumference(double radius)

{

return (2 \* M\_PI \* radius);

}

double getArea(double radius)

{

return (M\_PI \* radius \* radius);

}

1. C program to check prime, armstrong and perfect numbers using functions.

Takes more time to compete.

#include <stdio.h>

#include <math.h>

int isPrime(int num);

int isArmstrong(int num);

int isPerfect(int num);

int main()

{

int num;

printf("Enter any number: ");

scanf("%d", &num);

if(isPrime(num))

{

printf("%d is Prime number.\n", num);

}

else

{

printf("%d is not Prime number.\n", num);

}

if(isArmstrong(num))

{

printf("%d is Armstrong number.\n", num);

}

else

{

printf("%d is not Armstrong number.\n", num);

}

if(isPerfect(num))

{

printf("%d is Perfect number.\n", num);

}

else

{

printf("%d is not Perfect number.\n", num);

}

return 0;

}

int isPrime(int num)

{

int i;

for(i=2; i<=num/2; i++)

{

if(num%i == 0)

{

return 0;

}

}

return 1;

}

int isArmstrong(int num)

{

int lastDigit, sum, originalNum, digits;

sum = 0;

originalNum = num;

digits = (int) log10(num) + 1;

while(num > 0)

{

lastDigit = num % 10;

sum = sum + round(pow(lastDigit, digits));

num = num / 10;

}

return (originalNum == sum);

}

int isPerfect(int num)

{

int i, sum, n;

sum = 0;

n = num;

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

{

if(n%i == 0)

{

sum += i;

}

}

return (num == sum);

}

1. C program to add two number using pointers.

#include <stdio.h>

int main()

{

int num1, num2, sum;

int \*ptr1, \*ptr2;

ptr1 = &num1;

ptr2 = &num2;

printf("Enter any two numbers: ");

scanf("%d%d", ptr1, ptr2);

sum = \*ptr1 + \*ptr2;

printf("Sum = %d", sum);

return 0;

}

1. Swap 2 numbers using Call by Value AND Call by reference.

Call by reference

#include <stdio.h>

void swap(int \* num1, int \* num2);

int main()

{

int num1, num2;

printf("Enter two numbers: ");

scanf("%d%d", &num1, &num2);

printf("Before swapping in main n");

printf("Value of num1 = %d \n", num1);

printf("Value of num2 = %d \n\n", num2);

swap(&num1, &num2);

printf("After swapping in main n");

printf("Value of num1 = %d \n", num1);

printf("Value of num2 = %d \n\n", num2);

return 0;

}

void swap(int \* num1, int \* num2)

{

int temp;

temp = \*num1;

\*num1= \*num2;

\*num2= temp;

printf("After swapping in swap function n");

printf("Value of num1 = %d \n", \*num1);

printf("Value of num2 = %d \n\n", \*num2);

}

Call by value(time need go read)

#include<stdio.h>

void swap(int,int);

void main( )

{

int n1,n2;

printf("Enter the two numbers to be swapped\n");

scanf("%d%d",&n1,&n2);

printf("\nThe values of n1 and n2 in the main function before calling the swap function are n1=%d n2=%d",n1,n2);

swap(n1,n2);

printf("\nThe values of n1 and n2 in the main function after calling the swap function are n1=%d n2=%d",n1,n2);}

void swap(int n1,int n2)

{

int temp;

temp=n1;

n1=n2;

n2=temp;

printf("\nThe values of n1 and n2 in the swap function after swapping are n1=%d n2=%d",n1,n2);

}

1. C program to copy an array to another array AND reverse an array using pointers.(samasya gambhir hai)

#include <stdio.h>

#define MAX\_SIZE 100

void printArr(int \*arr, int size);

int main()

{

int arr[MAX\_SIZE];

int size;

int \*left = arr;

int \*right;

printf("Enter size of array: ");

scanf("%d", &size);

right = &arr[size - 1];

printf("Enter elements in array: ");

while(left <= right)

{

scanf("%d", left++);

}

printf("\nArray before reverse: ");

printArr(arr, size);

left = arr;

while(left < right)

{

\*left ^= \*right;

\*right ^= \*left;

\*left ^= \*right;

left++;

right--;

}

printf("\nArray after reverse: ");

printArr(arr, size);

return 0;

}

void printArr(int \* arr, int size)

{

int \* arrEnd = (arr + size - 1);

while(arr <= arrEnd)

{

printf("%d, ", \*arr);

arr++;

}

}

***PATTERNS***

1. Square Star pattern

#include <stdio.h>

int main()

{

int i, j, N;

printf("Enter number of rows: ");

scanf("%d", &N);

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

{

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

{

printf("\*");

}

printf("\n");

}

return 0;

}

1. HOLLOW SQUARE PATTERN

#include <stdio.h>

int main()

{

int i, j, N;

printf("Enter number of rows: ");

scanf("%d", &N);

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

{

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

{

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

{

printf("\*");

}

else

{

printf(" ");

}

}

printf("\n");

}

return 0;

}

1. Hollow square pattern with diagonal

#include <stdio.h>

int main()

{

int i, j, N;

printf("Enter number of rows: ");

scanf("%d", &N);

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

{

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

{

if(i==1 || i==N || j==1 || j==N || i==j || j==(N - i + 1))

{

printf("\*");

}

else

{

printf(" ");

}

}

printf("\n");

}

return 0;

}

1. RHOMBUS STAR PATTERN

#include <stdio.h>

int main()

{

int i, j, rows;

printf("Enter rows: ");

scanf("%d", &rows);

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

{

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

{

printf(" ");

}

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

{

printf("\*");

}

printf("\n");

}

return 0;

}

1. Hollow rhombus pattern

#include <stdio.h>

int main()

{

int i, j, rows;

printf("Enter rows : ");

scanf("%d", &rows);

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

{

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

{

printf(" ");

}

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

{

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

printf("\*");

else

printf(" ");

}

printf("\n");

}

return 0;

}