KAPIL GUPTA SECTION-I BCA (AI AND DS)

C-PROGRAMMING ASSIGNMENT

1. WAP for hello world or this is my first c program.

#include <stdio.h>

int main()

{

printf("kapil gupta\n"); printf("Hello World");

return 0;

}



1. #include <stdio.h> WAP to add two numbers .

int main() {

int number1, number2, sum; printf("Enter two integers: ");

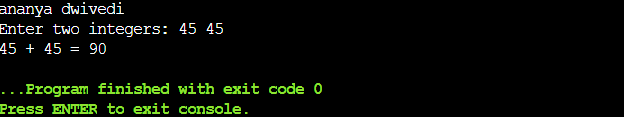
scanf("%d %d", &number1, &number2);

// calculate the sum

sum = number1 + number2;

printf("%d + %d = %d", number1, number2, sum); return 0;

}



1. WAP to find area of circle.

#include <stdio.h>

int main() {

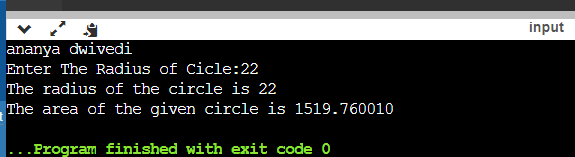
printf("kapil gupta\n"); float pie = 3.14;

int radius;

printf("Enter The Radius of Cicle:"); scanf("%d",&radius);

printf("The radius of the circle is %d\n" , radius); float area = (float)(pie\* radius \* radius); printf("The area of the given circle is %f", area); return 0;

}



1. WAP to divide two numbers.

#include<stdio.h> int main()

{

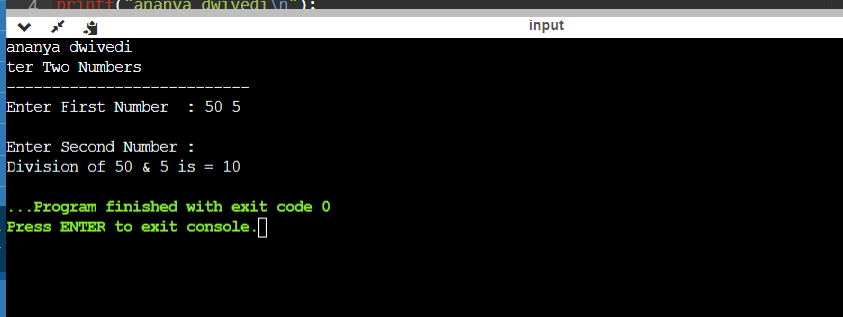
int num1,num2,div; Printf(“kapil gupta\n”); printf("\Enter Two Numbers\n"); printf(" \n");

printf("Enter First Number : "); scanf("%d", &num1); printf("\nEnter Second Number : "); scanf("%d",&num2); div=num1/num2;

printf("\nDivision of %d & %d is = %d",num1,num2,div);

return 0;

}



1. WAP to print ASCII value.

#include <stdio.h>

int main() {

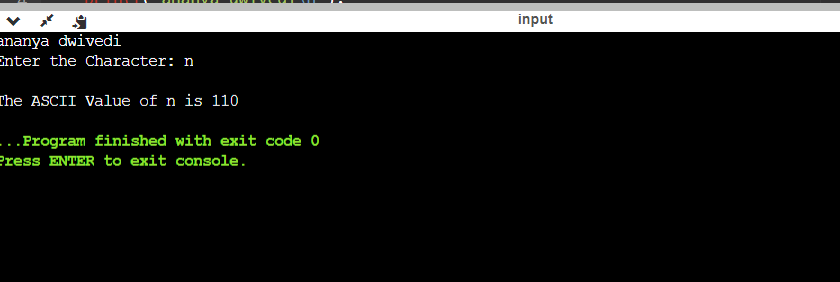
printf("kapil gupta\n"); char ch;

// assigning a letter to ch printf("Enter the Character: "); scanf("%c", & ch);

// displaying the ASCII value of the letter stored in ch printf("\nThe ASCII Value of %c is %d", ch, ch);

return 0;

}



1. WAP to multiply floating point numbers.

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

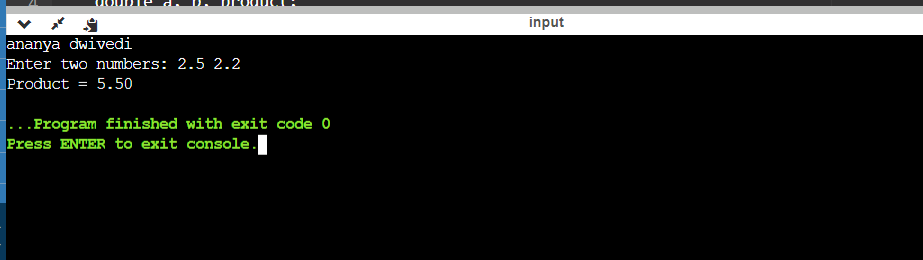
printf("kapil gupta\n"); double a, b, product; printf("Enter two numbers: "); scanf("%lf %lf", &a, &b);

// Calculating product product = a \* b;

// %.2lf displays number up to 2 decimal point printf("Product = %.2lf", product);

return 0;

}



1. WAP to swap two variables number by using third variable.

#include <stdio.h> int main()

{

printf("kapil gupta\n"); int var1, var2, temp; printf("Enter two integersn"); scanf("%d%d", &var1, &var2);

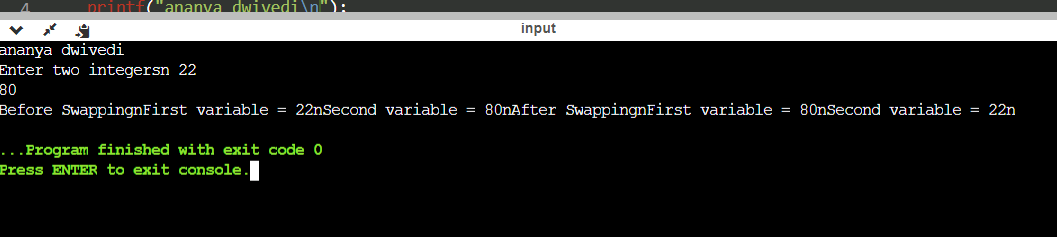
printf("Before SwappingnFirst variable = %dnSecond variable =

%dn", var1, var2); temp = var1;

var1 = var2; var2 = temp;

printf("After SwappingnFirst variable = %dnSecond variable =

%dn", var1, var2); return 0;

}

1. write a c program to swap two numbers without using third variable.

#include<stdio.h> int main()

{

printf("kapil gupta\n"); int x, y;

printf("Input value for x & y: \n"); scanf("%d%d",&x,&y);

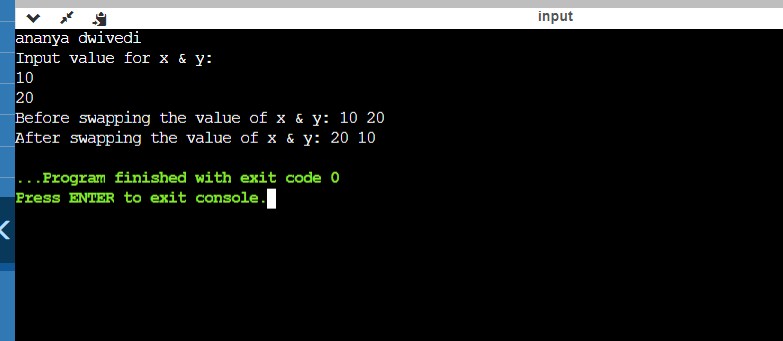
printf("Before swapping the value of x & y: %d %d",x,y); x=x+y;

y=x-y;

x=x-y;

printf("\nAfter swapping the value of x & y: %d %d",x,y); return 0;

}



1. write a c program to swap three numbers without using third variable.

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

printf("kapil gupta\n"); int a,b,c;

printf(" Enter values of a, b and c \n"); scanf("%d %d %d",&a,&b,&c);

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

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

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

b=a-b-c;

c=a-b-c;

a=a-b-c;

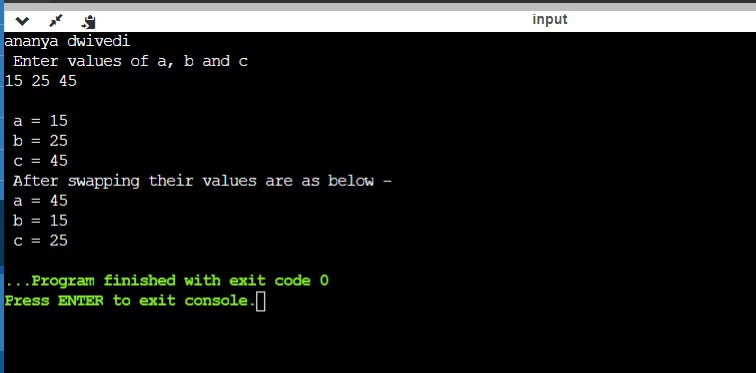
printf("\n After swapping their values are as below -");

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

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

printf("\n c = %d",c); return 0;

}



1. WAP to find area of rectangle. #include <stdio.h>

int main()

{

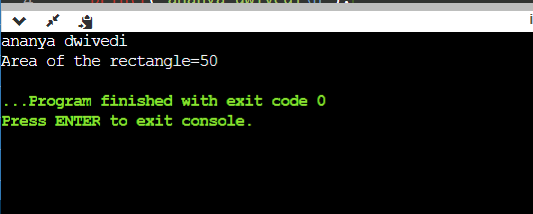
printf("kapil gupta\n"); int width=5;

int height=10;

int area=width\*height;

printf("Area of the rectangle=%d",area);

}



1. WAP to find area of square.

#include <stdio.h> int main()

{

printf("kapil gupta\n"); int s=13;

int area\_square=s\*s;

printf("Area of the square=%d",area\_square);

}



1. WAP to find area of right angled triangle,isoceles triangle,any triangle with 3 sides.

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

// Function to calculate the area of a right-angled triangle float area\_of\_right\_triangle(float base, float height) {

return 0.5 \* base \* height;

}

// Function to calculate the area of an isosceles triangle float area\_of\_isosceles\_triangle(float base, float side) { float height = sqrt(side \* side - (base/2) \* (base/2));

return 0.5 \* base \* height;

}

// Function to calculate the area of any triangle using Heron's formula

float area\_of\_any\_triangle(float a, float b, float c) { float s = (a + b + c) / 2; // Calculate semi-perimeter return sqrt(s \* (s - a) \* (s - b) \* (s - c));

}

int main() {

float base, height, side, a, b, c;

// Input for right-angled triangle printf("kapil gupta\n");

printf("Enter the base of the right-angled triangle: "); scanf("%f", &base);

printf("Enter the height of the right-angled triangle: "); scanf("%f", &height);

float area\_right = area\_of\_right\_triangle(base, height); printf("Area of the right-angled triangle: %f\n", area\_right);

// Input for isosceles triangle

printf("Enter the base of the isosceles triangle: "); scanf("%f", &base);

printf("Enter one of the equal sides of the isosceles triangle: ");

scanf("%f", &side);

float area\_isosceles = area\_of\_isosceles\_triangle(base, side); printf("Area of the isosceles triangle: %f\n", area\_isosceles);

// Input for any triangle

printf("Enter the length of side 'a': "); scanf("%f", &a);

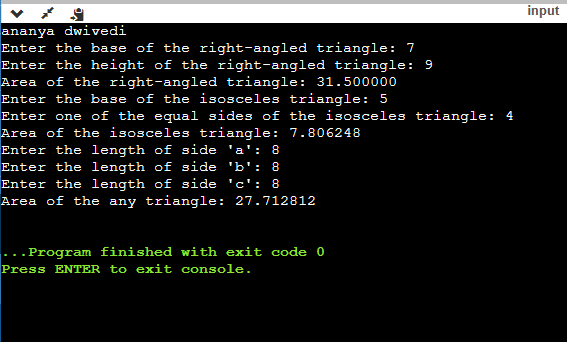
printf("Enter the length of side 'b': "); scanf("%f", &b);

printf("Enter the length of side 'c': "); scanf("%f", &c);

float area\_any = area\_of\_any\_triangle(a, b, c); printf("Area of the any triangle: %f\n", area\_any);

return 0;

}



1. WAP to find area and volume of cube.

#include<stdio.h>

int main(){

printf("kapil gupta\n");

float a;

float surface\_area,volume;

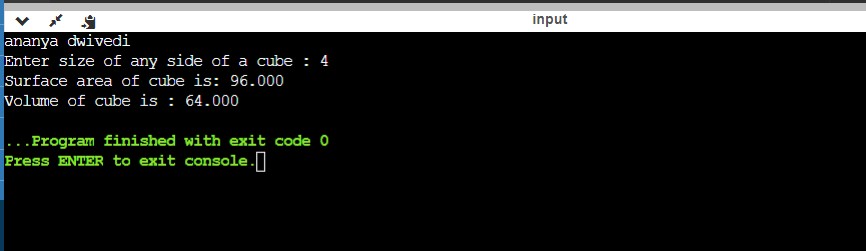
printf("Enter size of any side of a cube : "); scanf("%f",&a);

surface\_area = 6 \* (a \* a); volume = a \* a \* a;

printf("Surface area of cube is: %.3f",surface\_area); printf("\nVolume of cube is : %.3f",volume);

return 0;

}



1. WAP to find area and volume of cuboid.

#include <stdio.h>

// Function to calculate the surface area of a cuboid

float surface\_area\_of\_cuboid(float length, float width, float height) {

return 2 \* (length \* width + length \* height + width \* height);

}

// Function to calculate the volume of a cuboid

float volume\_of\_cuboid(float length, float width, float height) { return length \* width \* height;

}

int main() {

float length, width, height;

// Input for the dimensions of the cuboid printf("kapil gupta\n");

printf("Enter the length of the cuboid: "); scanf("%f", &length);

printf("Enter the width of the cuboid: "); scanf("%f", &width);

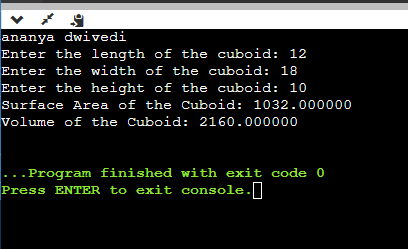
printf("Enter the height of the cuboid: "); scanf("%f", &height);

float surface\_area = surface\_area\_of\_cuboid(length, width, height);

float volume = volume\_of\_cuboid(length, width, height); printf("Surface Area of the Cuboid: %f\n", surface\_area); printf("Volume of the Cuboid: %f\n", volume);

return 0;

}



1. WAP to find the largest number using the logical AND operator.

#include <stdio.h> int main ()

{

printf("kapil gupta\n");

// declare integer type variable int x, y, z;

printf (" Enter the first number: "); scanf ("%d", &x);

printf (" Enter the second number: "); scanf ("%d", &y);

printf (" Enter the third number: "); scanf ("%d", &z);

// use logical AND operator to validate the condition if ( x >= y && x >= z )

{

printf (" %d is the largest number of all. ", x);

}

else if ( y >= x && y >= z)

{

printf (" %d is the largest number of all. ", y);

}

else

{

printf ( " %d is the largest number of all. ", z);

}

return 0;

}



1. WAP to validate the username and password entered by the user is correct or not using the predefined username and password.

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

int main() {

// Predefined username and password

char predefinedUsername[] = "myusername"; char predefinedPassword[] = "mypassword";

char enteredUsername[100]; char enteredPassword[100];

// Input username and password from the user printf("kapil gupta\n");

printf("Enter username: "); scanf("%s", enteredUsername); printf("Enter password: "); scanf("%s", enteredPassword);

// Compare entered username and password with predefined values

if (strcmp(enteredUsername, predefinedUsername) == 0 && strcmp(enteredPassword, predefinedPassword) == 0) {

printf("Login successful! Welcome, %s\n", enteredUsername);

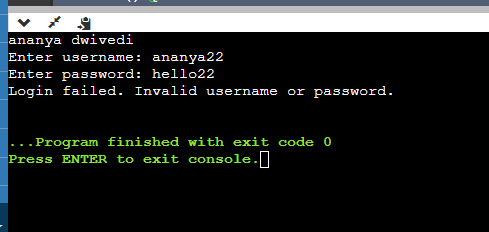
} else {

printf("Login failed. Invalid username or password.\n");

}

return 0;

}



1. WAP to input the positive number from the user to perform the left shift operator.

#include <stdio.h> int main ()

{

printf("kapil gupta\n");

// dec#lare local variable int num;

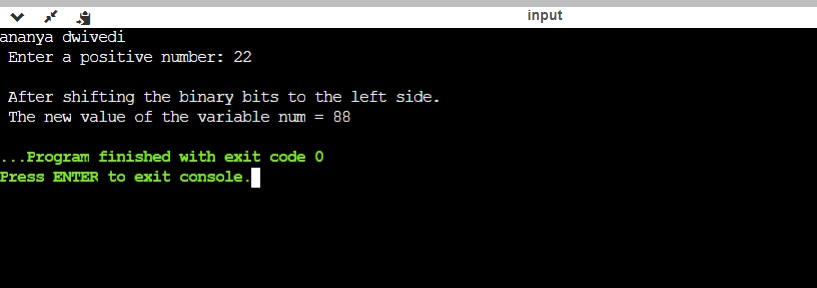
printf (" Enter a positive number: "); scanf (" %d", &num);

// use left shift operator to shift the bits

num = (num << 2); // It shifts two bits at the left side printf (" \n After shifting the binary bits to the left side. ");

printf (" \n The new value of the variable num = %d", num); return 0;

}



1. WAP to input the positive number from the user to perform the right shift operator.

#include <stdio.h> int main ()

{

printf("kapil gupta\n");

// declare local variable int num, bit;

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

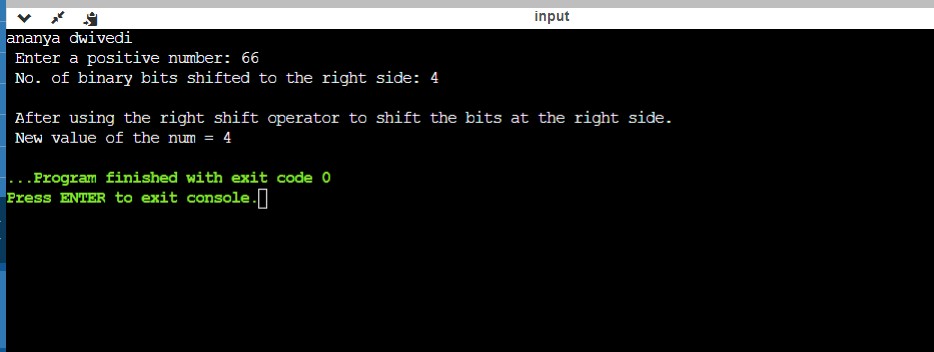
printf (" No. of binary bits shifted to the right side: "); scanf (" %d", &bit);

// use right shift operator to shift the bits num = (num >> bit);

printf (" \n After using the right shift operator to shift the bits at the right side. ");

printf (" \n New value of the num = %d", num); return 0;

}



1. WAP for pre and post decrement operator on two integers and print both original value and updated value.

# include <stdio.h>

int main() { int a = 10; int b = 5;

// Pre-decrement operator printf("kapil gupta");

printf("Using pre-decrement operator (--a and --b):\n"); printf("Original value of 'a': %d\n", a);

printf("Original value of 'b': %d\n", b);

--a;

--b;

printf("Updated value of 'a': %d\n", a); printf("Updated value of 'b': %d\n", b);

// Reset values a = 10;

b = 5;

// Post-decrement operator

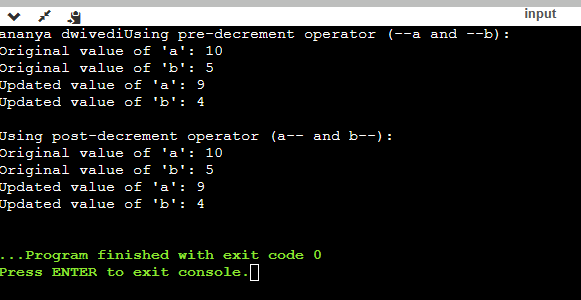
printf("\nUsing post-decrement operator (a-- and b--):\n"); printf("Original value of 'a': %d\n", a);

printf("Original value of 'b': %d\n", b); a--;

b--;

printf("Updated value of 'a': %d\n", a); printf("Updated value of 'b': %d\n", b);

return 0;



1. WAP to perform the post increment and post decrement operator on two integers and print both original value and updated value.

#include <stdio.h>

int main() {

int num1 = 5, num2 = 10;

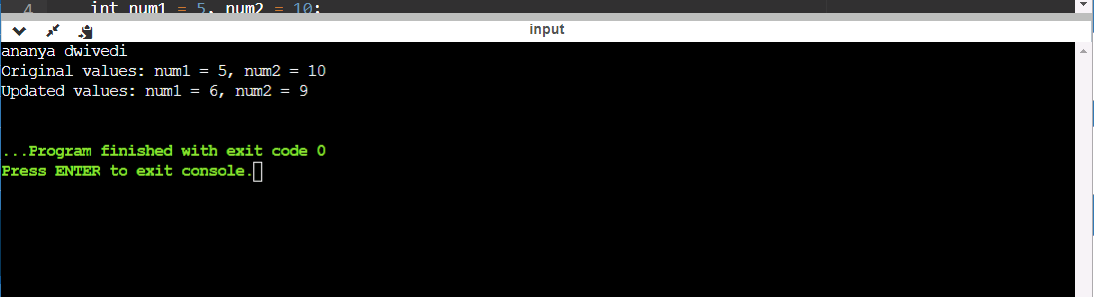
printf("Original values: num1 = %d, num2 = %d\n", num1, num2);

num1++; num2--;

printf("Updated values: num1 = %d, num2 = %d\n", num1, num2);

return 0;

}



1. for an integer number and to check whether it is divisible by 9 or WAP 7 using OR logical operator.

#include <stdio.h>

int main() { int num;

printf("kapil gupta\n");

printf("Enter an integer number: "); scanf("%d", &num);

if (num % 9 == 0 || num % 7 == 0) {

printf("%d is divisible by 9 or 7.\n", num);

} else {

printf("%d is not divisible by 9 or 7.\n", num);

}

return 0;

}



1. WAP to identify gender in single character and print full gender (Ex: if input is 'M' or 'm' – it should print "Male").

#include <stdio.h>

int main() { char gender;

printf("kapil gupta\n");

printf("Enter a single character (M/F): "); scanf(" %c", &gender);

if (gender == 'M' || gender == 'm') { printf("Full gender: Male\n");

} else if (gender == 'F' || gender == 'f') {

printf("Full gender: Female\n");

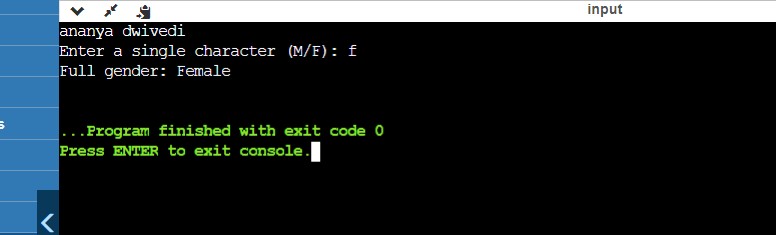
} else {

printf("Invalid input\n");

}

return 0;

}



1. WAP to print all natural numbers in reverse(from n to 1). #include <stdio.h>

int main() { int n;

printf("kapil gupta\n");

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

printf("Natural numbers in reverse order from %d to 1:\n", n);

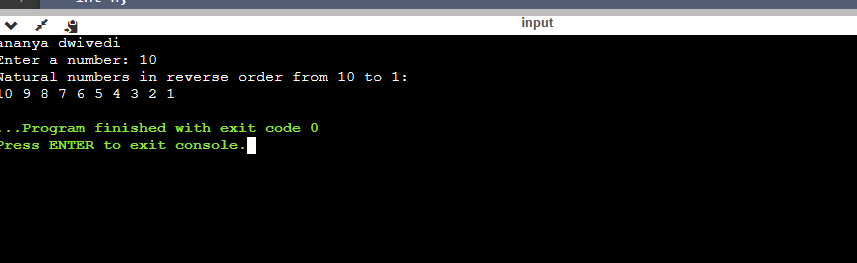
for (int i = n; i >= 1; i--) {

printf("%d ", i);

}

return 0;

}



1. WAP to print all alphabets from a to z.

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

char alphabet; printf("kapil gupta\n");

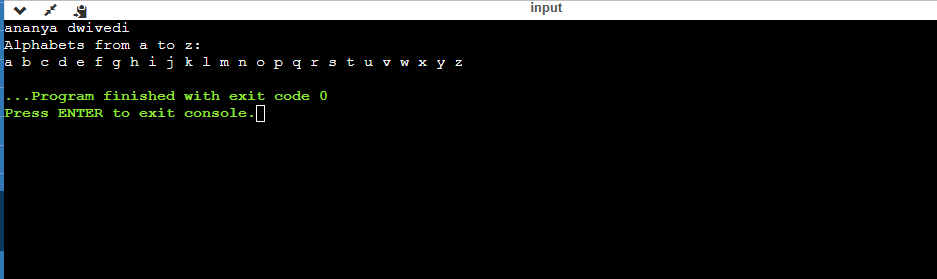
printf("Alphabets from a to z:\n");

for (alphabet = 'a'; alphabet <= 'z'; alphabet++) { printf("%c ", alphabet);

}

return 0;

}



1. WAP to print all natural numbers from 1 to n.

#include <stdio.h>

int main() { int n;

printf("kapil gupta\n");

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

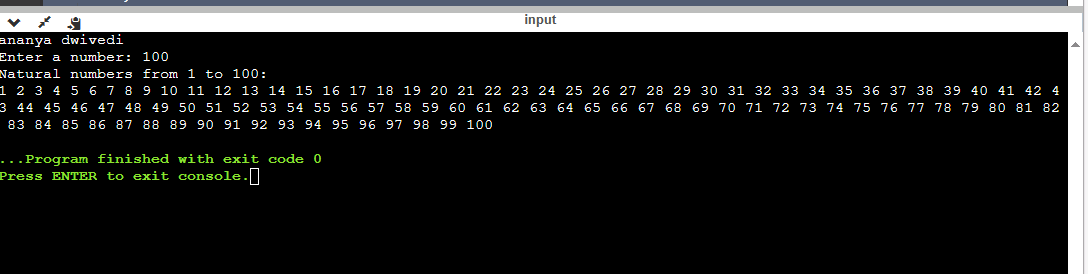
printf("Natural numbers from 1 to %d:\n", n);

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

}

return 0;

}



1. to print WAP all even numbers between 1 to 100.

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

printf("kapil gupta\n");

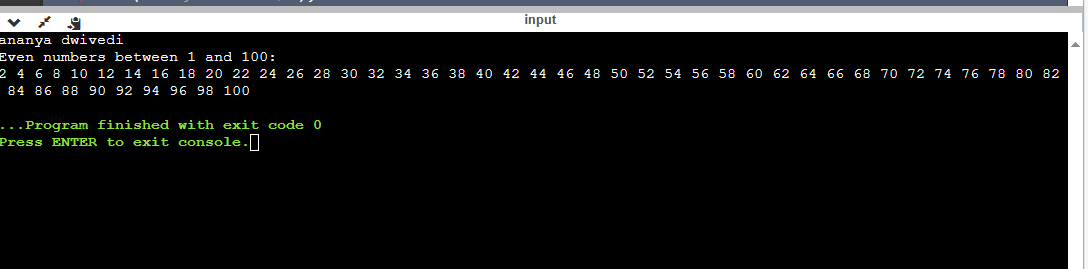
printf("Even numbers between 1 and 100:\n");

for (int i = 2; i <= 100; i += 2) { printf("%d ", i);

}

return 0;

}



1. WAP to print all the odd numbers between 1 to 100.

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

printf("kapil gupta");

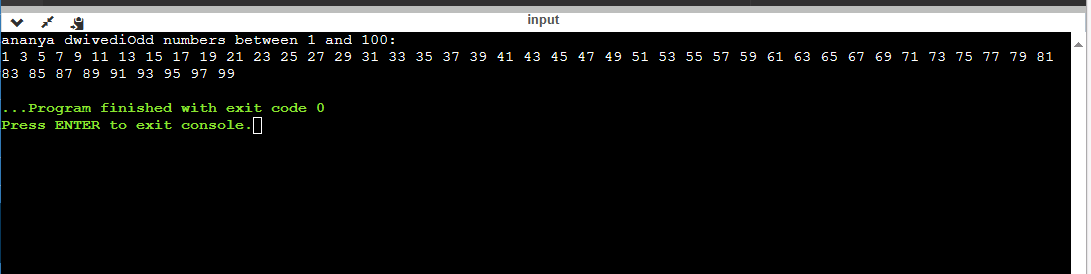
printf("Odd numbers between 1 and 100:\n");

for (int i = 1; i <= 100; i += 2) { printf("%d ", i);

}

return 0;

}



1. WAP to find the sum of all natural numbers from 1 to n.

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

int n, sum; printf("kapil gupta\n");

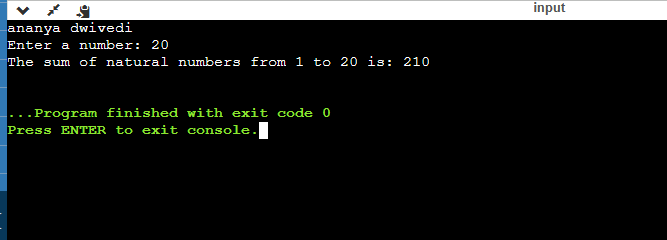
printf("Enter a number: "); scanf("%d", &n);

sum = (n \* (n + 1)) / 2;

printf("The sum of natural numbers from 1 to %d is: %d\n", n, sum);

return 0;

}



1. WAP to find the sum of all even numbers from 1 to n.

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

int n, sum; printf("kapil gupta\n");

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

sum = 0;

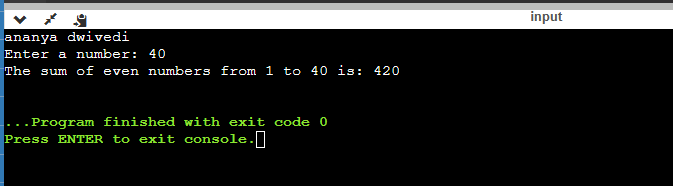
for (int i = 2; i <= n; i += 2) { sum += i;

}

printf("The sum of even numbers from 1 to %d is: %d\n", n, sum);

return 0;

}



1. WAP to find the sum of all odd numbers from 1 to n.

#include <stdio.h>

int main() { int n, sum;

printf("kapil gupta\n");

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

sum = 0;

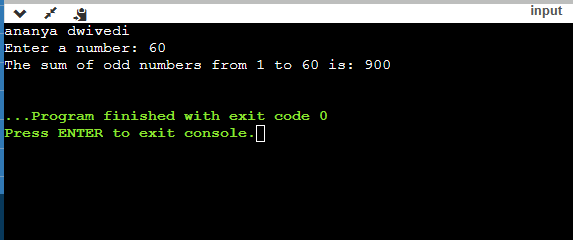
for (int i = 1; i <= n; i += 2) { sum += i;

}

printf("The sum of odd numbers from 1 to %d is: %d\n", n, sum);

return 0;

}



1. WAP to print multiplication table of any number. #include <stdio.h>

int main() { int number;

printf("kapil gupta\n");

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

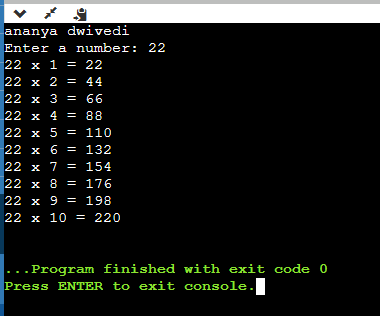
for (int i = 1; i <= 10; i++) {

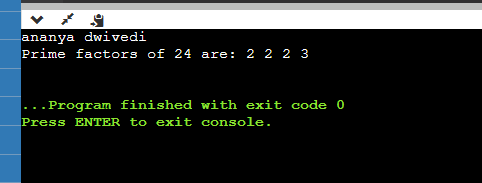
printf("%d x %d = %d\n", number, i, number \* i);

}

return 0;

}





1. WAP to check whether a number is armstrong number or not.

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

int main()

{

int originalNum, num, lastDigit, digits, sum;

/\* Input number from user \*/ printf("kapil gupta\n");

printf("Enter any number to check Armstrong number: "); scanf("%d", &num);

sum = 0;

/\* Copy the value of num for processing \*/ originalNum = num;

/\* Find total digits in num \*/

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

/\* Calculate sum of power of digits \*/ while(num > 0)

{

/\* Extract the last digit \*/ lastDigit = num % 10;

/\* Compute sum of power of last digit \*/ sum = sum + round(pow(lastDigit, digits));

/\* Remove the last digit \*/ num = num / 10;

}

/\* Check for Armstrong number \*/ if(originalNum == sum)

{

printf("%d is ARMSTRONG NUMBER", originalNum);

}

else

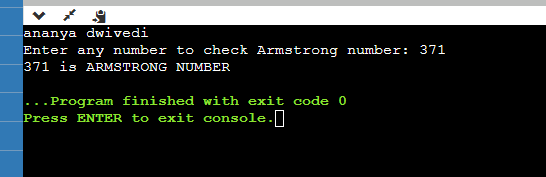
{

printf("%d is NOT ARMSTRONG NUMBER", originalNum);

}

return 0;

}



1. WAP to find Armstrong numbers between 1 to n.

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

int main()

{

int num, lastDigit, digits, sum, i, end;

/\* Input upper limit from user \*/

printf("Enter upper limit: "); scanf("%d", &end);

printf("Armstrong number between 1 to %d are: \n", end); printf("kapil gupta\n");

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

{

sum = 0;

/\* Copy the value of num for processing \*/ num = i;

/\* Find total digits in num \*/ digits = (int) log10(num) + 1;

/\* Calculate sum of power of digits \*/ while(num > 0)

{

/\* Extract last digit \*/ lastDigit = num % 10;

pow()

}

// Find sum of power of digits

// Use ceil() function to overcome any rounding errors by sum = sum + ceil(pow(lastDigit, digits));

/\* Remove the last digit \*/ num = num / 10;

/\* Check for Armstrong number \*/ if(i == sum)

{

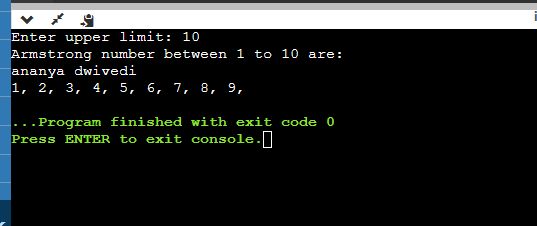
printf("%d, ", i);

}

}

return 0;

}



1. WAP to check whether a number is perfect number or not.

#include <stdio.h>

int main()

{

int i, num, sum = 0;

/\* Input a number from user \*/ printf("kapil gupta\n");

printf("Enter any number to check perfect number: "); scanf("%d", &num);

/\* Calculate sum of all proper divisors \*/ for(i = 1; i <= num / 2; i++)

{

/\* If i is a divisor of num \*/ if(num%i == 0)

{

sum += i;

}

}

/\* Check whether the sum of proper divisors is equal to num

\*/

if(sum == num && num > 0)

{

printf("%d is PERFECT NUMBER", num);

}

else

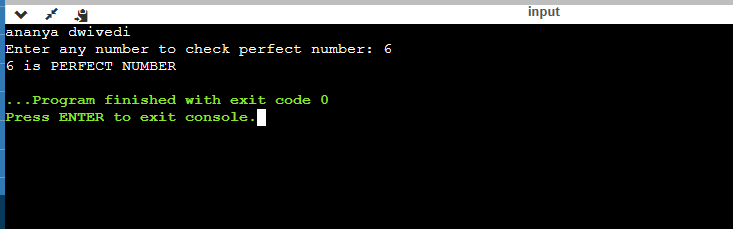
{

printf("%d is NOT PERFECT NUMBER", num);

}

return 0;

}



1. WAP to find perfect numbers between 1 to n. include <stdio.h>

int main()

{

int i, j, end, sum;

/\* Input upper limit to print perfect number \*/ printf("kapil gupta\n");

printf("Enter upper limit: "); scanf("%d", &end);

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

/\* Iterate from 1 to end \*/ for(i=1; i<=end; i++)

{

sum = 0;

/\* Check whether the current number i is Perfect number or not \*/

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

{

if(i % j == 0)

{

sum += j;

}

}

/\* If the current number i is Perfect number \*/ if(sum == i)

{

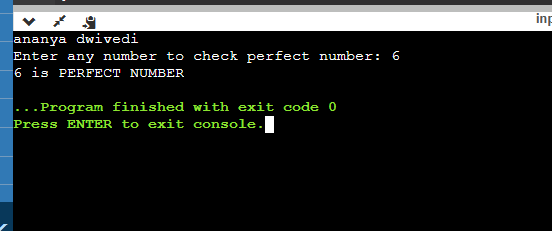
printf("%d, ", i);

}

}

return 0;

}



1. WAP to check whether a number is strong or not.

#include <stdio.h>

int factorial(int num) { int fact = 1;

while (num > 0) {

fact \*= num; num--;

}

return fact;

}

int isStrongNumber(int num) { int originalNum = num;

int sum = 0; while (num > 0) {

int digit = num % 10; sum += factorial(digit); num /= 10;

}

if (sum == originalNum) return 1;

else

return 0;

}

int main() { int num;

printf("kapil gupta\n"); printf("Enter a number: "); scanf("%d", &num);

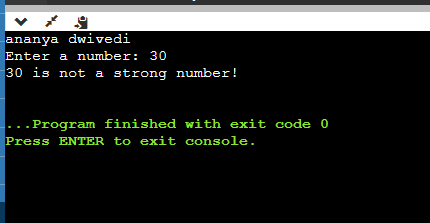
if (isStrongNumber(num))

printf("%d is a strong number!\n", num); else

printf("%d is not a strong number!\n", num);

return 0;

}



1. WAP to print strong numbers between 1 to n.

#include <stdio.h> int main()

{

int i, j, cur, lastDigit, end;

long long fact, sum;

/\* Input upper limit from user \*/ printf("Enter upper limit: "); scanf("%d", &end);

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

/\* Iterate from 1 to end \*/ for(i=1; i<=end; i++)

{

/\* Number to check for strong number \*/ cur = i;

sum = 0;

/\* Find the sum of factorial of digits \*/ while(cur > 0)

{

fact = 1ll;

lastDigit = cur % 10;

/\* Find factorial of last digit of current num. \*/ for( j=1; j<=lastDigit; j++)

{

fact = fact \* j;

}

sum += fact;

cur /= 10;

}

/\* Print 'i' if it is strong number \*/ if(sum == i)

{

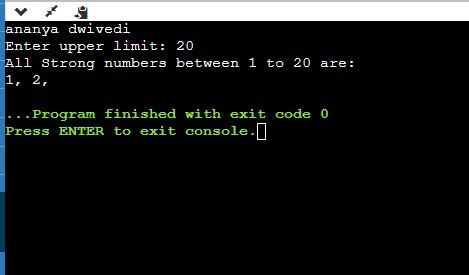
printf("%d, ", i);

}

}

return 0;

}



1. WAP to print fibonacci series upto n terms.

#include <stdio.h>

int main()

{

int a, b, c, i, terms;

/\* Input number from user \*/ printf("kapil gupta\n"); printf("Enter number of terms: "); scanf("%d", &terms);

/\* Fibonacci magic initialization \*/ a = 0;

b = 1;

c = 0;

printf("Fibonacci terms: \n");

/\* Iterate through n terms \*/ for(i=1; i<=terms; i++)

{

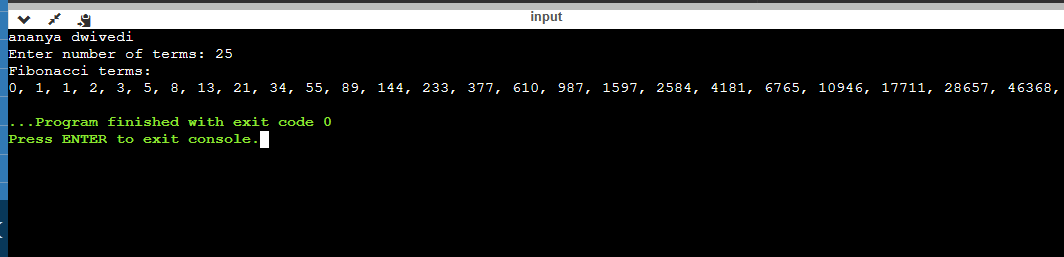
printf("%d, ", c);

a = b; // Copy n-1 to n-2

b = c; // Copy current to n-1 c = a + b; // New term

}

return 0;

}

1. WAP to find ones complement of a binary number.

#include <stdio.h>

#include <string.h>

void onesComplement(char binary[]) { int length = strlen(binary);

for (int i = 0; i < length; i++) { if (binary[i] == '0')

binary[i] = '1';

else if (binary[i] == '1') binary[i] = '0';

}

printf("kapil gupta\n");

printf("One's complement of %s is: %s\n", binary, binary);

}

int main() {

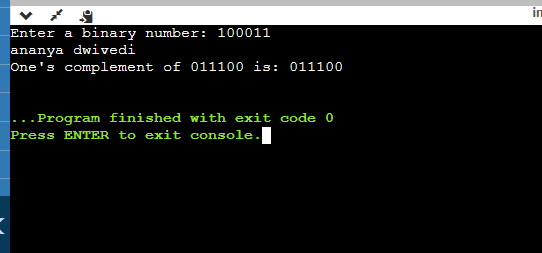
char binary[100];

printf("Enter a binary number: ");

scanf("%s", binary); onesComplement(binary);

return 0;

}



1. WAP to find twos complement of a binary number.

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

void twosComplement(char binary[]) { printf("kapil gupta\n");

int length = strlen(binary); int carry = 1;

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

if (binary[i] == '0' && carry == 1) { binary[i] = '1';

carry = 0;

}

else if (binary[i] == '1' && carry == 1) { binary[i] = '0';

}

}

printf("Two's complement of %s is: %s\n", binary, binary);

}

int main() {

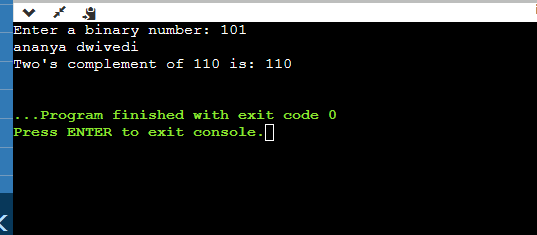
char binary[100];

printf("Enter a binary number: "); scanf("%s", binary);

twosComplement(binary);

return 0;

}



1. WAP to covert binary into octal number system .

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

int binaryToOctal(long long binaryNumber) { printf("kapil gupta\n");

int octalNumber = 0, decimalNumber = 0, i = 0;

while (binaryNumber != 0) {

decimalNumber += (binaryNumber % 10) \* pow(2, i);

++i;

binaryNumber /= 10;

}

i = 1;

while (decimalNumber != 0) {

octalNumber += (decimalNumber % 8) \* i; decimalNumber /= 8;

i \*= 10;

}

return octalNumber;

}

int main() {

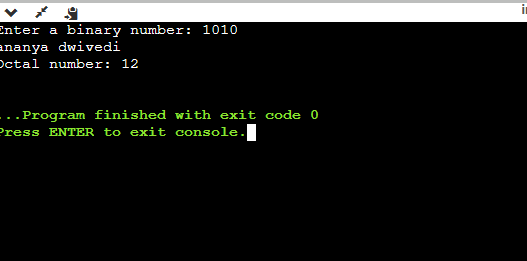
long long binaryNumber;

printf("Enter a binary number: "); scanf("%lld", &binaryNumber);

int octalNumber = binaryToOctal(binaryNumber); printf("Octal number: %d\n", octalNumber);

return 0;

}



1. WAP to convert binary to decimal number system. #include <stdio.h>

#include <math.h>

int binaryToDecimal(long long binaryNumber) { int decimalNumber = 0, i = 0, remainder;

while (binaryNumber != 0) { remainder = binaryNumber % 10;

decimalNumber += remainder \* pow(2, i); binaryNumber /= 10;

++i;

}

return decimalNumber;

}

int main() {

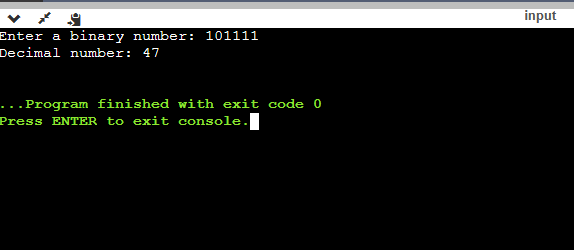
long long binaryNumber;

printf("Enter a binary number: "); scanf("%lld", &binaryNumber);

int decimalNumber = binaryToDecimal(binaryNumber); printf("Decimal number: %d\n", decimalNumber);

return 0;

}



1. WAP to convert binary to hexadecimal number system.

#include <stdlib.h> #include <string.h>

// Function to convert binary to hexadecimal char\* binaryToHex(char\* binary) {

printf("kapil gupta\n");

int decimal = strtol(binary, NULL, 2);

char\* hex = (char\*)malloc(10 \* sizeof(char)); sprintf(hex, "%X", decimal);

return hex;

}

int main() {

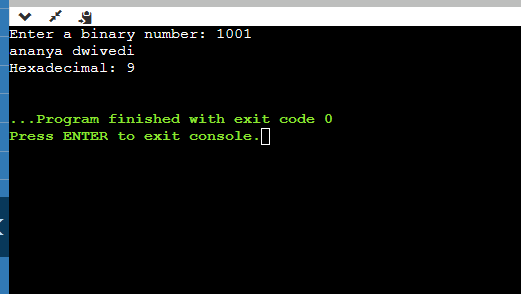
char binary[100];

printf("Enter a binary number: "); scanf("%s", binary);

char\* hex = binaryToHex(binary); printf("Hexadecimal: %s\n", hex);

free(hex); return 0;

}



1. WAP to convert octal to binary number.

#include <stdio.h>

int main(){

printf("kapil gupta\n");

int OCTALVALUES[] = {0, 1, 10, 11, 100, 101, 110, 111};

long long octal, tempOctal, binary, place; int rem;

/\* Input Octal number from user \*/ printf("Enter any Octal number: "); scanf("%lld", &octal);

tempOctal = octal; binary = 0;

place = 1;

/\* Convert octal to binary \*/ while(tempOctal > 0)

{

/\* Extract the last digit of octal \*/ rem = tempOctal % 10;

/\*

* Get the binary equivalent of octal digit
* add it to the binary variable

\*/

binary = (OCTALVALUES[rem] \* place) + binary;

/\* Remove the last octal digit \*/ tempOctal /= 10;

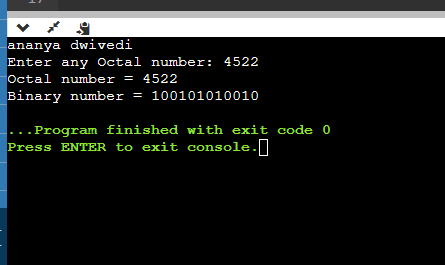
/\* Increase the place value \*/ place \*= 1000;

}

printf("Octal number = %lld\n", octal); printf("Binary number = %lld", binary);

return 0;

}



1. WAP to convert octal to decimal number system.

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

int main()

{

long long octal, tempOctal, decimal; int rem, place;

/\* Input octal number from user \*/ printf("kapil gupta\n"); printf("Enter any octal number: "); scanf("%lld", &octal);

tempOctal = octal;

decimal = 0;

place = 0;

/\*

\* Convert octal to decimal

\*/

while(tempOctal > 0)

{

/\* Extract the last digit of octal \*/ rem = tempOctal % 10;

/\* Convert last octal digit to decimal \*/ decimal += pow(8, place) \* rem;

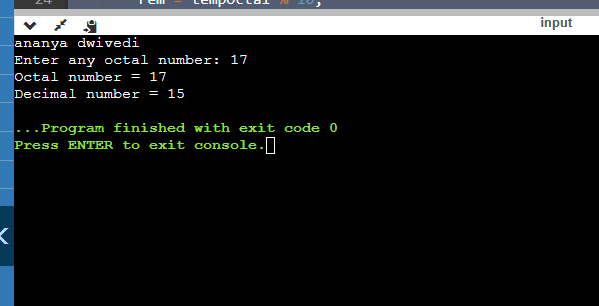
/\* Remove the last octal digit \*/ tempOctal /= 10;

place++;

}

printf("Octal number = %lld\n", octal); printf("Decimal number = %lld", decimal); return 0;

}



1. WAP to convert octal to hexadecimal number system. #include <stdio.h>

#include <string.h>

// Function to convert an octal digit to its decimal equivalent int octalToDecimal(char octalDigit) {

return octalDigit - '0';

}

// Function to convert a decimal digit to its hexadecimal equivalent

char decimalToHex(int decimalDigit) {

if (decimalDigit >= 0 && decimalDigit <= 9) {

return (char)(decimalDigit + '0');

} else {

// For decimal digits 10 to 15, use A to F for hexadecimal return (char)(decimalDigit - 10 + 'A');

}

}

int main() {

char octal[20]; printf("kapil gupta\n");

printf("Enter an octal number: "); scanf("%s", octal);

int octalLength = strlen(octal); char hexadecimal[20];

int decimal = 0;

int decimalPlaceValue = 1;

// Convert the octal number to decimal for (int i = octalLength - 1; i >= 0; i--) {

decimal += octalToDecimal(octal[i]) \* decimalPlaceValue; decimalPlaceValue \*= 8;

}

int index = 0;

// Convert decimal to hexadecimal while (decimal > 0) {

int remainder = decimal % 16; hexadecimal[index++] = decimalToHex(remainder); decimal /= 16;

}

// Reverse the hexadecimal string for (int i = 0; i < index / 2; i++) {

char temp = hexadecimal[i]; hexadecimal[i] = hexadecimal[index - 1 - i]; hexadecimal[index - 1 - i] = temp;

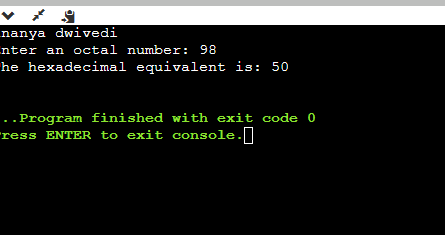
}

hexadecimal[index] = '\0';

printf("The hexadecimal equivalent is: %s\n", hexadecimal);

return 0;

}



1. WAP to convert decimal to binary number system.

#include <stdio.h>

int main()

{

long long decimal, tempDecimal, binary; int rem, place = 1;

binary = 0;

/\* Input decimal number from user \*/ printf("kapil gupta\n"); printf("Enter any decimal number: "); scanf("%lld", &decimal);

tempDecimal = decimal;

/\* Decimal to binary conversion \*/ while(tempDecimal > 0)

{

rem = tempDecimal % 2; binary = (rem \* place) + binary; tempDecimal /= 2;

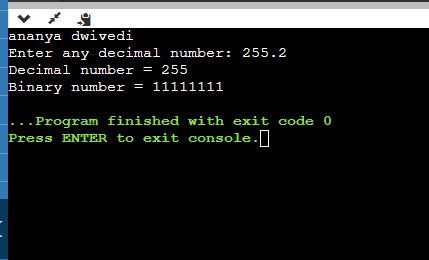
place \*= 10;

}

printf("Decimal number = %lld\n", decimal); printf("Binary number = %lld", binary);

return 0;

}



1. WAP to convert decimal to octal number system.

#include <stdio.h>

int main()

{

long long decimal, tempDecimal, octal; int i, rem, place = 1;

octal = 0;

/\* Input decimal number from user \*/ printf("kapil gupta\n"); printf("Enter any decimal number: "); scanf("%lld", &decimal);

tempDecimal = decimal;

/\* Decimal to octal conversion \*/ while(tempDecimal > 0)

{

rem = tempDecimal % 8; octal = (rem \* place) + octal; tempDecimal /= 8;

place \*= 10;

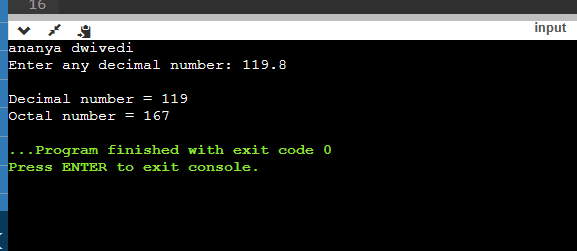
}

printf("\nDecimal number = %lld\n", decimal);

printf("Octal number = %lld", octal);

return 0;

}



1. WAP to convert decimal to hexadecimal number system.

#include <stdio.h> #include <stdlib.h>

// Function to convert decimal to hexadecimal void convertDecimalToHexadecimal(int decimal) {

printf("kapil gupta\n"); int quotient, remainder; char hexadecimal[100];

int i = 0;

quotient = decimal; while (quotient != 0) {

remainder = quotient % 16; if (remainder < 10)

hexadecimal[i++] = 48 + remainder; else

hexadecimal[i++] = 55 + remainder; quotient = quotient / 16;

}

printf("Hexadecimal: "); for (int j = i - 1; j >= 0; j--) {

printf("%c", hexadecimal[j]);

}

printf("\n");

}

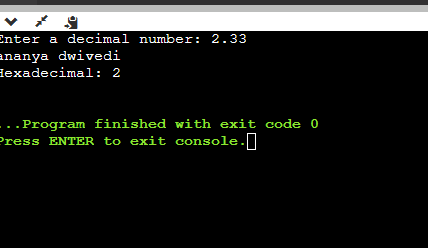
int main() { int decimal;

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

convertDecimalToHexadecimal(decimal);

return 0;

}



1. WAP to convert hexadecimal to binary number system.

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

// Function to convert a hexadecimal digit to its binary equivalent

char\* hexToBinary(char hexDigit) {

switch (hexDigit) {

case '0': return "0000";

case '1': return "0001";

case '2': return "0010";

case '3': return "0011";

case '4': return "0100";

case '5': return "0101";

case '6': return "0110";

case '7': return "0111";

case '8': return "1000";

case '9': return "1001"; case 'A': return "1010"; case 'B': return "1011"; case 'C': return "1100"; case 'D': return "1101"; case 'E': return "1110"; case 'F': return "1111";

default: return NULL; // Invalid hexadecimal digit

}

}

int main() {

char hexadecimal[20]; printf("kapil gupta\n");

printf("Enter a hexadecimal number: "); scanf("%s", hexadecimal);

int hexLength = strlen(hexadecimal);

printf("The binary equivalent is: "); for (int i = 0; i < hexLength; i++) {

char\* binary = hexToBinary(hexadecimal[i]); if (binary) {

printf("%s", binary);

} else {

printf("Invalid hexadecimal digit: %c\n", hexadecimal[i]); return 1; // Exit with an error code

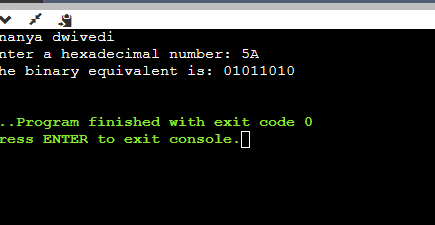
}

}

printf("\n");

return 0;

}



# PATTERN EXERCISES-

1. Star pattern programs - Write a C program to print the given star patterns.

\*

\*\*\*

\*\*\*\*\*

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

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

* 1. Pyramid Star Pattern

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

int rows, i, j, space; printf("kapil gupta\n");

printf("Enter the number of rows: "); scanf("%d", &rows);

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

for (space = 1; space <= rows - i; space++) { printf(" ");

}

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

printf("\*");

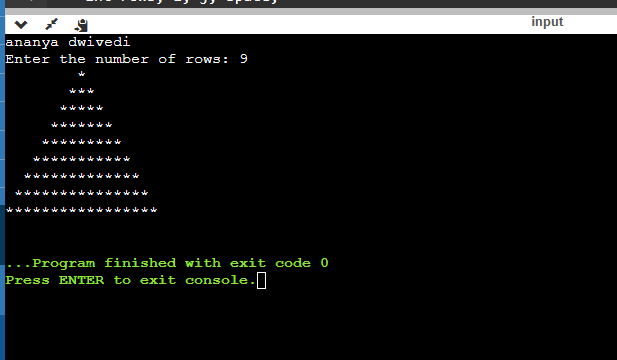
}

printf("\n");

}

return 0;

}



1. Hollow pyramid star patter.

#include <stdio.h>

int main() {

int rows, i, j, space; printf("kapil gupta\n");

printf("Enter the number of rows: "); scanf("%d", &rows);

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

for (space = 1; space <= rows - i; space++) { printf(" ");

}

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

if (i == rows || j == 1 || j == 2 \* i - 1) { printf("\*");

} else {

printf(" ");

}

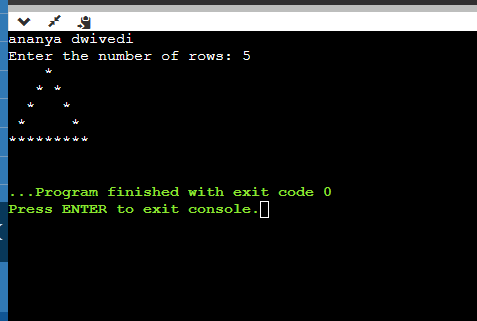
}

printf("\n");

}

return 0;

}



1. Inverted pyramid star pattern.

#include <stdio.h> int main()

{

int n,m=1;

printf("kapil gupta\n");

printf("Enter the number of rows"); scanf("%d",&n);

for(int i=n;i>=1;i--)

{

for(int j=1;j<m;j++)

{

printf(" ");

}

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

{

printf("\*");

}

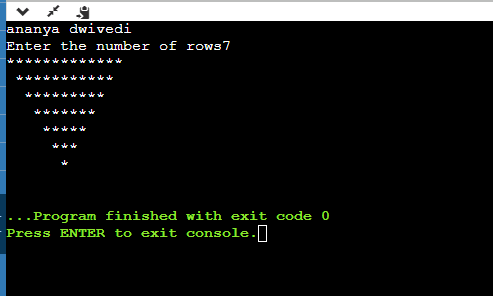
m++;

printf("\n");

}

return 0;

}



1. Hollow inverted pyramid star pattern.

#include <stdio.h>

int main()

{

int n,m=1;

printf("kapil gupta\n"); printf("Enter the number of rows"); scanf("%d",&n);

for(int i=n;i>=1;i--)

{

for(int j=1;j<m;j++)

{

printf(" ");

}

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

{

if(k==1 || k==2\*i-1 || i==n) printf("\*");

else printf(" ");

}

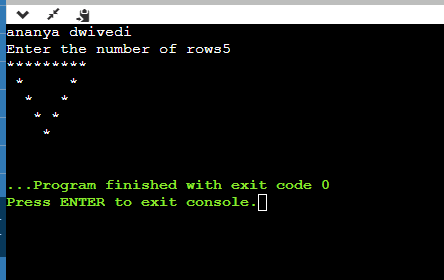
m++;

printf("\n");

}

return 0;

}



1. Half diamond star pattern.

#include<stdio.h> int main()

{

int i, j, rows; printf("kapil gupta");

printf("Enter Half Diamond Rows = "); scanf("%d", &rows);

printf("Half Diamond Star Pattern\n"); for(i = 1; i <= rows; i++)

{

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

{

printf("\*");

}

printf("\n");

}

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

{

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

{

printf("\*");

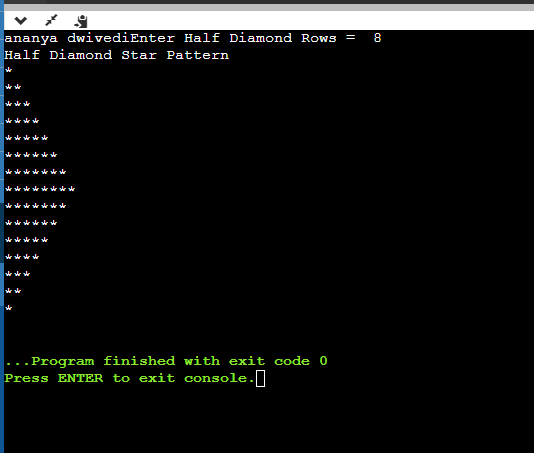
}

printf("\n");

}

return 0;

}



1. Mirrored Half diamond star pattern.

#include <stdio.h>

int main() { int rows;

printf("kapil gupta\n"); printf("Enter the number of rows: "); scanf("%d", &rows);

for (int i = 1; i <= rows; i++) { for (int j = 1; j <= i; j++) {

printf("\* ");

}

printf("\n");

}

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

printf("\* ");

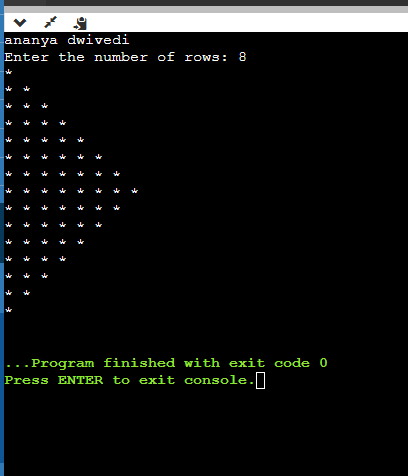
}

printf("\n");

}

return 0;

}



# NUMBER PATTERN PROGRAMS-

2.WAP to print the given number patterns-

1.Square number patterns.

11111

11111

11111

11111

11111

#include <stdio.h>

int main()

{

int rows, cols, i, j;

/\* Input rows and columns from user \*/ printf("kapil gupta\n"); printf("Enter number of rows: "); scanf("%d", &rows);

printf("Enter number of columns: "); scanf("%d", &cols);

/\* Iterate through rows \*/ for(i=1; i<=rows; i++)

{

/\* Iterate through columns \*/ for(j=1; j<=cols; j++)

{

printf("1");

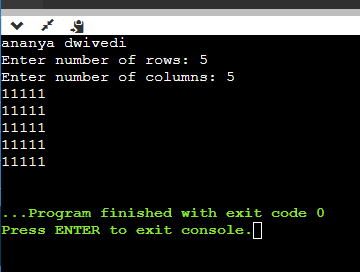
}

printf("\n");

}

return 0;

}



Number pattern 1. Number pattern 11111

00000

11111

00000

11111

int main()

{

int rows, cols, i, j;

/\* Input rows and columns from user \*/ printf("kapil gupta\n"); printf("Enter number of rows: "); scanf("%d", &rows);

printf("Enter number of columns: "); scanf("%d", &cols);

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

{

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

{

// Print 1 if current row is odd if(i%2 == 1)

{

printf("1");

}

else

{

printf("0");

}

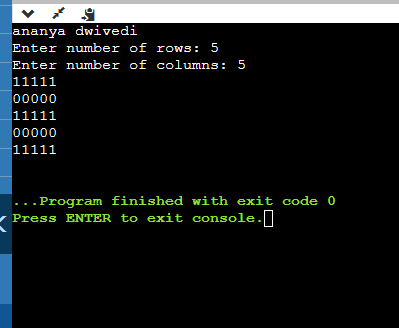
}

printf("\n");

}

return 0;

}



Number pattern 2

01010

01010

01010

01010

#include <stdio.h>

int main() {

printf("kapil gupta\n"); int rows = 4;

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

for (int j = 1; j <= rows; j++) { if (j % 2 == 0) {

printf("1");

} else {

printf("0");

}

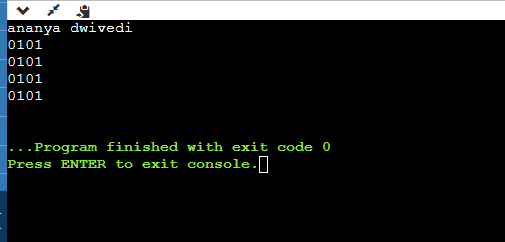
}

printf("\n");

}

return 0;

}



Number pattern 3.

11111

10001

10001

10001

11111

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

printf("kapil gupta\n"); int rows = 5;

for (int i = 1; i <= rows; i++) { for (int j = 1; j <= rows; j++) {

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

printf("1");

} else {

printf("0");

}

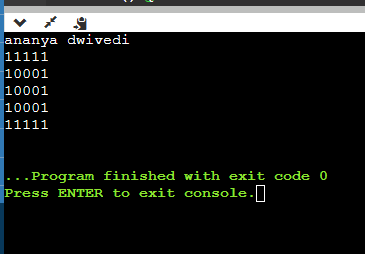
}

printf("\n");

}

return 0;

}



Number pattern 4

11111

11111

11011

11111

11111

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

printf("kapil gupta\n"); int rows = 5;

for (int i = 1; i <= rows; i++) { for (int j = 1; j <= rows; j++) {

if (i == 3 && j == 3) { printf("0");

} else {

printf("1");

}

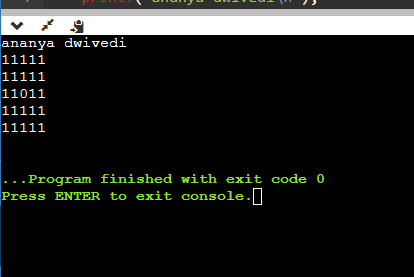
}

printf("\n");

}

return 0;

}



Number pattern 5

10101

01010

10101

01010

10101

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

printf("kapil gupta\n"); int rows = 5;

for (int i = 1; i <= rows; i++) { for (int j = 1; j <= rows; j++) {

if ((i + j) % 2 == 0) {

printf("1");

} else {

printf("0");

}

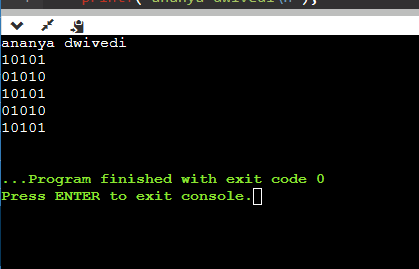
}

printf("\n");

}

return 0;

}



# If...Else Exercises

1. Write a C program to find maximum between two numbers.

#include <stdio.h>

int main()

{

int num1, num2;

/\* Input two numbers from user \*/ printf("kapil gupta\n"); printf("Enter two numbers: "); scanf("%d%d", &num1, &num2);

/\* If num1 is maximum \*/ if(num1 > num2)

{

printf("%d is maximum", num1);

}

/\* If num2 is maximum \*/ if(num2 > num1)

{

printf("%d is maximum", num2);

}

/\* Additional condition check for equality \*/ if(num1 == num2)

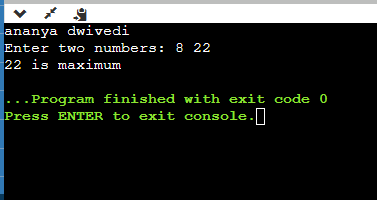
{

printf("Both are equal");

}

return 0;

}



# Write a C program to find maximum between three numbers.

# include <stdio.h>

int main()

{

int num1, num2, num3, max;

/\* Input three numbers from user \*/ printf("kapil gupta\n"); printf("Enter three numbers: ");

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

if(num1 > num2)

{

if(num1 > num3)

{

/\* If num1 > num2 and num1 > num3 \*/ max = num1;

}

else

{

/\* If num1 > num2 but num1 > num3 is not true \*/ max = num3;

}

}

else

{

if(num2 > num3)

{

/\* If num1 is not > num2 and num2 > num3 \*/ max = num2;

}

else

{

/\* If num1 is not > num2 and num2 > num3 \*/ max = num3;

}

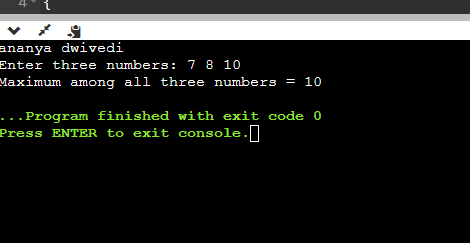
}

/\* Print maximum value \*/

printf("Maximum among all three numbers = %d", max);

return 0;

}



# Write a C program to check whether a number is negative, positive or zero.

#include <stdio.h> int main()

{

int num;

/\* Input number from user \*/ printf("kapil gupta\n"); printf("Enter any number: "); scanf("%d", &num);

if(num > 0)

{

printf("Number is POSITIVE");

}

if(num < 0)

{

printf("Number is NEGATIVE");

}

if(num == 0)

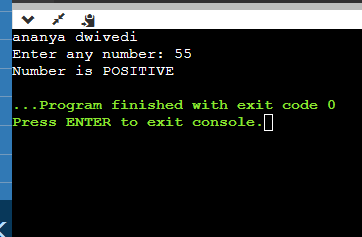
{

printf("Number is ZERO");

}

return 0;

}



# Write a C program to check whether a number is divisible by 5 and 11 or not.

#include <stdio.h>

int main()

{

int num;

/\* Input number from user \*/ printf("kapil gupta\n"); printf("Enter any number: "); scanf("%d", &num);

/\*

* If num modulo division 5 is 0
* and num modulo division 11 is 0 then
* the number is divisible by 5 and 11 both

\*/

if((num % 5 == 0) && (num % 11 == 0))

{

printf("Number is divisible by 5 and 11");

}

else

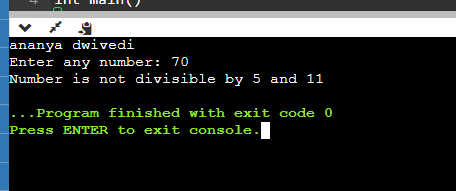
{

printf("Number is not divisible by 5 and 11");

}

return 0;

}



# Write a C program to check whether a number is even or odd.

#include <stdio.h>

int main()

{

int num;

/\* Input number from user \*/ printf("kapil gupta\n");

printf("Enter any number to check even or odd: "); scanf("%d", &num);

/\* Check if the number is divisible by 2 then it is even \*/ if(num % 2 == 0)

{

/\* num % 2 is 0 \*/ printf("Number is Even.");

}

else

{

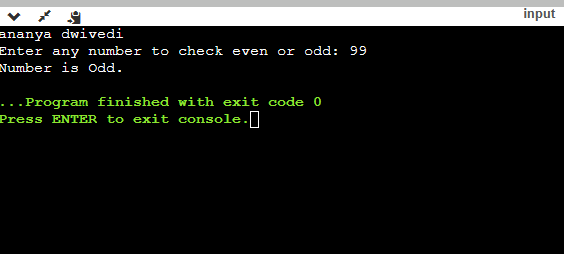
/\* num % 2 is 1 \*/

printf("Number is Odd.");

}

return 0;

}



# Write a C program to check whether a

year is leap year or not.

#include <stdio.h>

int main()

{

int year;

/\* Input year from user \*/ printf("kapil gupta\n"); printf("Enter year : "); scanf("%d", &year);

/\*

* If year is exactly divisible by 4 and year is not divisible by 100
* or year is exactly divisible by 400 then
* the year is leap year.
* Else year is normal year

\*/

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

{

printf("LEAP YEAR");

}

else

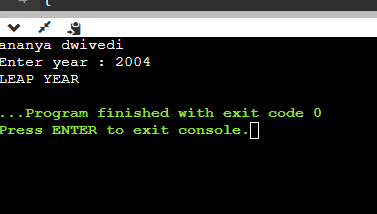
{

printf("COMMON YEAR");

}

return 0;

}



# Write a C program to check whether a character is alphabet or not.

#include <stdio.h> int main()

{

char ch;

/\* Input a character from user \*/ printf("kapil gupta\n"); printf("Enter any character: "); scanf("%c", &ch);

if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))

{

printf("Character is an ALPHABET.");

}

else

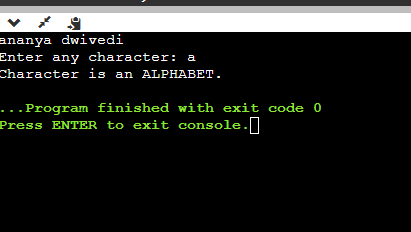
{

printf("Character is NOT ALPHABET.");

}

return 0;

}



# Write a C program to input any alphabet and check whether it is vowel or consonant

.

#include <stdio.h>

int main()

{

char ch;

/\* Input character from user \*/ printf("kapil gupta\n"); printf("Enter any character: "); scanf("%c", &ch);

/\* Condition for vowel \*/

if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' || ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')

{

printf("'%c' is Vowel.", ch);

}

else if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))

{

/\* Condition for consonant \*/ printf("'%c' is Consonant.", ch);

}

else

{

/\*

* If it is neither vowel nor consonant
* then it is not an alphabet.

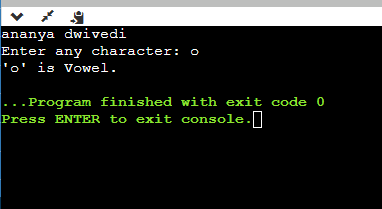
\*/

printf("'%c' is not an alphabet.", ch);

}

return 0;

}



# Write a C program to input any character and check whether it is alphabet, digit or special.

#include <stdio.h>

int main()

{

char ch;

/\* Input character from user \*/ printf("kapil gupta\n"); printf("Enter any character: "); scanf("%c", &ch);

/\* Alphabet check \*/

if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))

{

printf("'%c' is alphabet.", ch);

}

else if(ch >= '0' && ch <= '9')

{

printf("'%c' is digit.", ch);

}

else

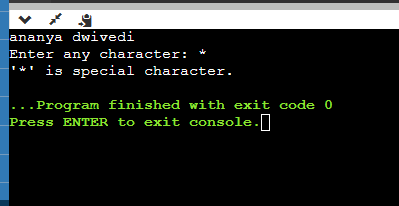
{

printf("'%c' is special character.", ch);

}

return 0;

}



# Write a C program to check whether a character is uppercase or lowercase alphabet.

#include <stdio.h>

int main()

{

char ch;

/\* Input character from user \*/ printf("kapil gupta\n"); printf("Enter any character: "); scanf("%c", &ch);

if(ch >= 'A' && ch <= 'Z')

{

printf("'%c' is uppercase alphabet.", ch);

}

else if(ch >= 'a' && ch <= 'z')

{

printf("'%c' is lowercase alphabet.", ch);

}

else

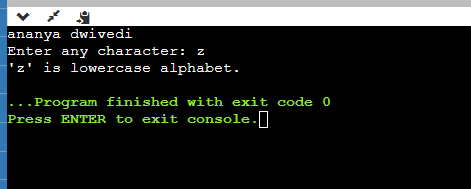
{

printf("'%c' is not an alphabet.", ch);

}

return 0;

}



# Write a C program to input week number and print week day.

#include <stdio.h>

int main()

{

int week;

/\* Input week number from user \*/ printf("kapil gupta"); printf("Enter week number (1-7): "); scanf("%d", &week);

if(week == 1)

{

printf("Monday");

}

else if(week == 2)

{

printf("Tuesday");

}

else if(week == 3)

{

printf("Wednesday");

}

else if(week == 4)

{

printf("Thursday");

}

else if(week == 5)

{

printf("Friday");

}

else if(week == 6)

{

printf("Saturday");

}

else if(week == 7)

{

printf("Sunday");

}

else

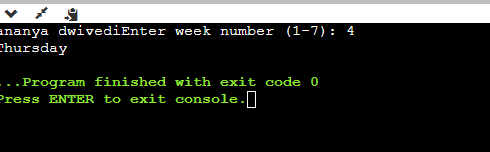
{

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

}

return 0;

}



# Write a C program to input month number and print number of days in that month.

#include <stdio.h>

int main()

{

int month;

/\* Input month number from user \*/ printf("Enter month number (1-12): "); printf("kapil gupta\n"); scanf("%d", &month);

if(month == 1)

{

printf("31 days");

}

else if(month == 2)

{

printf("28 or 29 days");

}

else if(month == 3)

{

printf("31 days");

}

else if(month == 4)

{

printf("30 days");

}

else if(month == 5)

{

printf("31 days");

}

else if(month == 6)

{

printf("30 days");

}

else if(month == 7)

{

printf("31 days");

}

else if(month == 8)

{

printf("31 days");

}

else if(month == 9)

{

printf("30 days");

}

else if(month == 10)

{

printf("31 days");

}

else if(month == 11)

{

printf("30 days");

}

else if(month == 12)

{

printf("31 days");

}

else

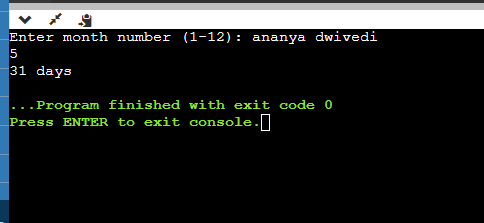
{

printf("Invalid input! Please enter month number between (1-12).");

}

return 0;

}



# Write a C program to count total number of notes in given amount.

#include <stdio.h>

int main()

{

int amount;

int note500, note100, note50, note20, note10, note5, note2, note1;

/\* Initialize all notes to 0 \*/

note500 = note100 = note50 = note20 = note10 = note5 = note2 = note1 = 0;

/\* Input amount from user \*/ printf("kapil gupta\n");

printf("Enter amount: "); scanf("%d", &amount);

if(amount >= 500)

{

note500 = amount/500; amount -= note500 \* 500;

}

if(amount >= 100)

{

note100 = amount/100; amount -= note100 \* 100;

}

if(amount >= 50)

{

note50 = amount/50; amount -= note50 \* 50;

}

if(amount >= 20)

{

note20 = amount/20; amount -= note20 \* 20;

}

if(amount >= 10)

{

note10 = amount/10; amount -= note10 \* 10;

}

if(amount >= 5)

{

note5 = amount/5; amount -= note5 \* 5;

}

if(amount >= 2)

{

note2 = amount /2; amount -= note2 \* 2;

}

if(amount >= 1)

{

note1 = amount;

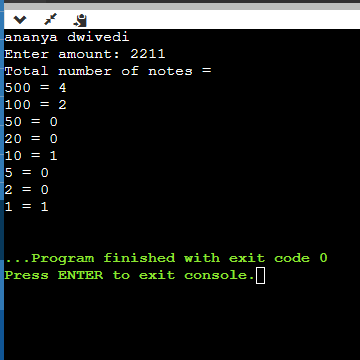
}

/\* Print required notes \*/ printf("Total number of notes = \n"); printf("500 = %d\n", note500); printf("100 = %d\n", note100); printf("50 = %d\n", note50); printf("20 = %d\n", note20); printf("10 = %d\n", note10); printf("5 = %d\n", note5);

printf("2 = %d\n", note2); printf("1 = %d\n", note1);

return 0;

}



# Write a C program to input month number and print number of days in that month.

#include <stdio.h> int main()

{

int month;

/\* Input month number from user \*/ printf("kapil gupta\n"); printf("Enter month number (1-12): ");

scanf("%d", &month);

if(month == 1)

{

printf("31 days");

}

else if(month == 2)

{

printf("28 or 29 days");

}

else if(month == 3)

{

printf("31 days");

}

else if(month == 4)

{

printf("30 days");

}

else if(month == 5)

{

printf("31 days");

}

else if(month == 6)

{

printf("30 days");

}

else if(month == 7)

{

printf("31 days");

}

else if(month == 8)

{

printf("31 days");

}

else if(month == 9)

{

printf("30 days");

}

else if(month == 10)

{

printf("31 days");

}

else if(month == 11)

{

printf("30 days");

}

else if(month == 12)

{

printf("31 days");

}

else

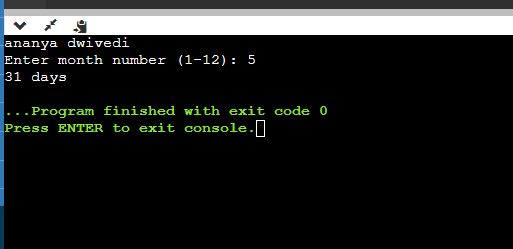
{

printf("Invalid input! Please enter month number between (1-12).");

}

return 0;

}



# Write a C program to count total number of notes in given amount.

#include <stdio.h> int main()

{

int amount;

int note500, note100, note50, note20, note10, note5, note2, note1;

/\* Initialize all notes to 0 \*/

note500 = note100 = note50 = note20 = note10 = note5 = note2 = note1 = 0;

/\* Input amount from user \*/ printf("kapil gupta\n"); printf("Enter amount: "); scanf("%d", &amount);

if(amount >= 500)

{

note500 = amount/500; amount -= note500 \* 500;

}

if(amount >= 100)

{

note100 = amount/100; amount -= note100 \* 100;

}

if(amount >= 50)

{

note50 = amount/50; amount -= note50 \* 50;

}

if(amount >= 20)

{

note20 = amount/20; amount -= note20 \* 20;

}

if(amount >= 10)

{

note10 = amount/10; amount -= note10 \* 10;

}

if(amount >= 5)

{

note5 = amount/5; amount -= note5 \* 5;

}

if(amount >= 2)

{

note2 = amount /2; amount -= note2 \* 2;

}

if(amount >= 1)

{

note1 = amount;

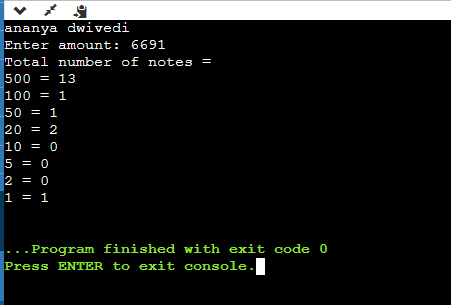
}

/\* Print required notes \*/ printf("Total number of notes = \n"); printf("500 = %d\n", note500); printf("100 = %d\n", note100); printf("50 = %d\n", note50); printf("20 = %d\n", note20); printf("10 = %d\n", note10); printf("5 = %d\n", note5);

printf("2 = %d\n", note2); printf("1 = %d\n", note1);

return 0;

}



# Write a C program to input angles of a triangle and check whether triangle is valid or not.

#include <stdio.h> int main()

{

int angle1, angle2, angle3, sum;

/\* Input all three angles of triangle \*/ printf("kapil gupta\n");

printf("Enter three angles of triangle: \n"); scanf("%d%d%d", &angle1, &angle2, &angle3);

/\* Calculate sum of angles \*/ sum = angle1 + angle2 + angle3;

/\*

* If sum of angles is 180 and
* angle1, angle2, angle3 is not 0 then
* triangle is valid.

\*/

if(sum == 180 && angle1 > 0 && angle2 > 0 && angle3 > 0)

{

printf("Triangle is valid.");

}

else

{

printf("Triangle is not valid.");

}

return 0;

}

# Write a C program to input all sides of a triangle and check whether triangle is valid or not.

#include <stdio.h>

int main()

{

int side1, side2, side3;

/\* Input sides of a triangle \*/ printf("kapil gupta\n"); printf("Enter three sides of triangle: ");

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

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

{

/\* If all sides are equal \*/ printf("Equilateral triangle.");

}

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

{

/\* If any two sides are equal \*/ printf("Isosceles triangle.");

}

else

{

/\* If none sides are equal \*/

printf("Scalene triangle.");

}

return 0;

}

# Write a C program to check whether the triangle is equilateral, isosceles or scalene triangle.

#include <stdio.h>

int main()

{

int side1, side2, side3;

/\* Input sides of a triangle \*/ printf("kapil gupta\n"); printf("Enter three sides of triangle: ");

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

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

{

/\* If all sides are equal \*/ printf("Equilateral triangle.");

}

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

{

/\* If any two sides are equal \*/ printf("Isosceles triangle.");

}

else

{

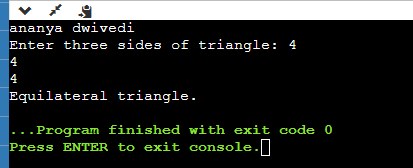
/\* If none sides are equal \*/

printf("Scalene triangle.");

}

return 0;

}



# Write a C program to find all roots of a quadratic equation.

#include <stdio.h>

#include <math.h> /\* Used for sqrt() \*/ int main()

{

float a, b, c;

float root1, root2, imaginary; float discriminant; printf("kapil gupta\n");

printf("Enter values of a, b, c of quadratic equation (aX^2 +

bX + c): ");

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

/\* Find discriminant of the equation \*/ discriminant = (b \* b) - (4 \* a \* c);

/\* Find the nature of discriminant \*/ if(discriminant > 0)

{

root1 = (-b + sqrt(discriminant)) / (2\*a); root2 = (-b - sqrt(discriminant)) / (2\*a);

printf("Two distinct and real roots exists: %.2f and %.2f", root1, root2);

}

else if(discriminant == 0)

{

root1 = root2 = -b / (2 \* a);

printf("Two equal and real roots exists: %.2f and %.2f", root1, root2);

}

else if(discriminant < 0)

{

root1 = root2 = -b / (2 \* a);

imaginary = sqrt(-discriminant) / (2 \* a);

printf("Two distinct complex roots exists: %.2f + i%.2f and

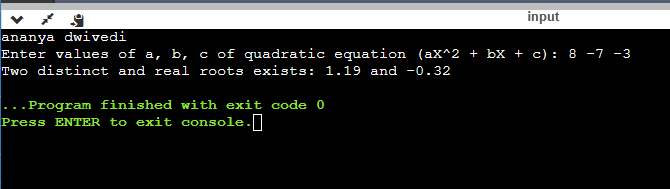
%.2f - i%.2f",

root1, imaginary, root2, imaginary);

}

return 0;

}



# Write a C program to calculate profit or loss.

#include <stdio.h> int main()

{

int cp,sp, amt;

/\* Input cost price and selling price of a product \*/ printf("kapil gupta\n ");

printf("Enter cost price: "); scanf("%d", &cp); printf("Enter selling price: "); scanf("%d", &sp);

if(sp > cp)

{

/\* Calculate Profit \*/ amt = sp - cp;

printf("Profit = %d", amt);

}

else if(cp > sp)

{

/\* Calculate Loss \*/ amt = cp - sp;

printf("Loss = %d", amt);

}

else

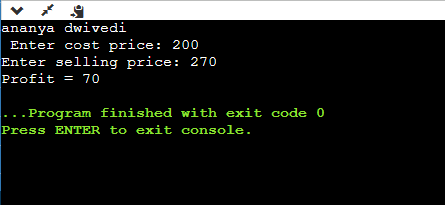
{

/\* Neither profit nor loss \*/ printf("No Profit No Loss.");

}

return 0;

}



# Write a C program to input marks of five subjects Physics, Chemistry, Biology,

Mathematics

and Computer. Calculate percentage and grade according to following:

Percentage >= 90% : Grade A Percentage >= 80% : Grade B Percentage >= 70% : Grade C

Percentage >= 60% : Grade D Percentage >= 40% : Grade E Percentage < 40% : Grade

#include <stdio.h>

int main()

{

int phy, chem, bio, math, comp; float per;

/\* Input marks of five subjects from user \*/ printf("kapil gupta\n");

printf("Enter five subjects marks: ");

scanf("%d%d%d%d%d", &phy, &chem, &bio, &math, &comp);

/\* Calculate percentage \*/

per = (phy + chem + bio + math + comp) / 5.0;

printf("Percentage = %.2f\n", per);

/\* Find grade according to the percentage \*/ if(per >= 90)

{

printf("Grade A");

}

else if(per >= 80)

{

printf("Grade B");

}

else if(per >= 70)

{

printf("Grade C");

}

else if(per >= 60)

{

printf("Grade D");

}

else if(per >= 40)

{

printf("Grade E");

}

else

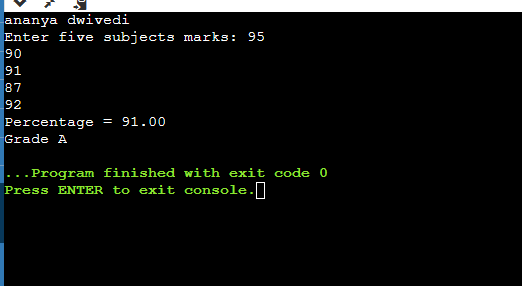
{

printf("Grade F");

}

return 0;

}



# Write a C program to input basic salary of an employee and calculate its Gross salary according

to following:

Basic Salary <= 10000 : HRA = 20%, DA = 80%

Basic Salary <= 20000 : HRA = 25%, DA =

90%

Basic Salary > 20000 : HRA = 30%, DA = 95

#include <stdio.h>

int main()

{

float basic, gross, da, hra;

/\* Input basic salary of employee \*/ printf("kapil gupta\n");

printf("Enter basic salary of an employee: "); scanf("%f", &basic);

/\* Calculate D.A and H.R.A according to specified conditions

\*/

if(basic <= 10000)

{

da = basic \* 0.8; hra = basic \* 0.2;

}

else if(basic <= 20000)

{

da = basic \* 0.9; hra = basic \* 0.25;

}

else

{

da = basic \* 0.95; hra = basic \* 0.3;

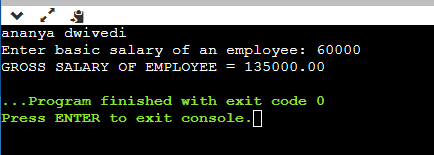
}

/\* Calculate gross salary \*/ gross = basic + hra + da;

printf("GROSS SALARY OF EMPLOYEE = %.2f", gross);

return 0;

}



# Write a C program to input electricity unit charges and calculate total electricity bill according

to the given condition:

For first 50 units Rs. 0.50/unit For next 100 units Rs. 0.75/unit For next 100 units Rs. 1.20/unit For unit above 250 Rs. 1.50/unit

An additional surcharge of 20% is added to the bil

#include <stdio.h>

int main()

{

int unit;

float amt, total\_amt, sur\_charge;

/\* Input unit consumed from user \*/ printf("kapil gupta\n"); printf("Enter total units consumed: "); scanf("%d", &unit);

/\* Calculate electricity bill according to given conditions \*/ 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);

}

/\*

* Calculate total electricity bill
* after adding surcharge

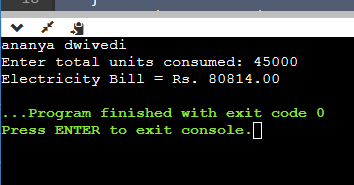
\*/

sur\_charge = amt \* 0.20; total\_amt = amt + sur\_charge;

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

return 0;

}



# Write a C program to convert specified days into years, weeks and days. #include

int main() { int days, years, weeks; days = 1329; // Converts days to years, weeks and 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 \n", days); return 0;}

#include <stdio.h>

int main()

{

printf("kapil gupta\n"); int days, years, weeks;

days = 1329; // Total number of days

// Converts days to years, weeks and days years = days/365; // Calculate years

weeks = (days % 365)/7; // Calculate weeks

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

// Print the results printf("Years: %d\n", years); printf("Weeks: %d\n", weeks); printf("Days: %d \n", days);

return 0;

}

