Code for Armstrong number:

Input: #include <stdio.h>

#include<math.h>

int main() {

int num,originalnum,remainder,count=0;

double result=0.0;

printf("enter number");

scanf("%d",&num);

originalnum=num;

while(num>0){

num/=10;

count++;

}

num=originalnum;

while(num>0){

remainder=num%10;

result+=pow(remainder,count);

num/=10;

}

if(result==originalnum){

printf("%d is a armstrong number",originalnum);

}

else{

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

}

return 0;

}

Output: enter number 407

407 is a armstrong number

Code to find the HCF of two numbers:

Input: #include <stdio.h>

int main() {

int num1, num2, i, hcf;

printf("Enter two numbers: ");

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

int smaller=(num1<num2)?num1:num2;

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

if(num1 % i == 0 && num2 % i == 0) {

hcf = i;

}

}

// Output the HCF

printf("HCF of %d and %d is: %d\n", num1, num2, hcf);

return 0;

}

Output:Enter two numbers: 30

60

HCF of 30 and 60 is: 30

Code to subtract two numbers by using bitwise operator:-

Input: #include <stdio.h>

int main() {

int a,b;

printf("enter a :");

scanf("%d",&a);

printf("enter b:");

scanf("%d",&b);

while(b!=0){

int borrow=(~a)&b;

a=a^b;

b=borrow<<1;

}

printf("subtraction of two numbers is %d",a);

return 0;

}

Output: enter a :5

enter b:6

subtraction of two numbers is -1

Code to swap two numbers by using four different methodes:-

1st method:

Input: #include <stdio.h>

int main() {

int a,b,temp;

printf("enter a:");

scanf("%d",&a);

printf("enter b:");

scanf("%d",&b);

printf("before swapping the numbers a=%d,b=%d\n",a,b);

temp=b;

b=a;

a=temp;

printf("after swapping the number a=%d,b=%d",a,b);

return 0;

}

Ouput: enter a:5

enter b:6

before swapping the numbers a=5,b=6

after swapping the number a=6,b=5

2nd method

Input:

#include <stdio.h>

int main() {

int a,b;

printf("enter a:");

scanf("%d",&a);

printf("enter b:");

scanf("%d",&b);

printf("before swapping the numbers a=%d,b=%d\n",a,b);

a=a+b;

b=a-b;

a=a-b;

printf("after swapping the numbers a=%d,b=%d",a,b);

return 0;

}

Output: enter a:7

enter b:8

before swapping the numbers a=7,b=8

after swapping the numbers a=8,b=7

3rd method

Input:

#include <stdio.h>

int main() {

int a,b;

printf("enter a:");

scanf("%d",&a);

printf("enter b:");

scanf("%d",&b);

printf("before swapping the numbers a=%d,b=%d\n",a,b);

b=(a+b)-(a=b);

printf("after swapping the numbers a=%d,b=%d",a,b);

return 0;

}

Ouput: enter a:7

enter b:8

before swapping the numbers a=7,b=8

after swapping the numbers a=8,b=7

4th method

Code:

Input: #include <stdio.h>

int main() {

int a,b;

printf("enter a:");

scanf("%d",&a);

printf("enter b:");

scanf("%d",&b);

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

int \*ptrA;

int \*ptrB;

ptrA=&a;

ptrB=&b;

int temp = \*ptrA;

\*ptrA = \*ptrB;

\*ptrB = temp;

printf("After swapping: a = %d, b = %d\n", \*ptrA, \*ptrB);

return 0;

}

Output: enter a:5

enter b:6

before swapping a=5,b=6

After swapping: a = 6, b = 5

Code for perfect number:

Input:

#include <stdio.h>

int main() {

int a,sum=0;

printf("enter a:");

scanf("%d",&a);

for(int i=1;i<=a/2;i++){

if(a%i==0){

sum+=i;

}

}

if(sum==a){

printf("%d is a perfect number",a);

}

else{

printf("%d is not a perfect number",a);

}

return 0;

}

Ouput:

enter a:6

6 is a perfect number

Code to find the quadrant:-

Input: #include <stdio.h>

int main() {

int n1,n2;

printf("enter n1:");

scanf("%d",&n1);

printf("enter n2:");

scanf("%d",&n2);

if(n1>0&&n2>0){

printf("first quadrant");

}

else if(n1<0&&n2>0){

printf("second quadrant");

}

else if(n1<0&&n2<0){

printf("third quadrant");

}

else if(n1>0&&n2<0){

printf("fourth quadrant");

}

else{

printf("origin");

}

return 0;

}

Output: enter n1:5

enter n2:-6

fourth quadrant

Code to convert binary to decimal and vice versa:-

Input: #include <stdio.h>

int main() {

int number,choice;

printf("1.binary to decimal\n");

printf("2. decimal to binary\n");

printf("enter a choice:");

scanf("%d",&choice);

if(choice==1){

printf("enter a binary:");

scanf("%d",&number);

int base=1,decimal=0,lastdigit;

while(number>0){

lastdigit=number%10;

number=number/10;

decimal+=lastdigit\*base;

base=base\*2;

}

printf("decimal:\n",decimal);

}

else if(choice==2){

printf("enter a decimal:");

scanf("%d",&number);

int binary[32];

int index=0;

while(number>0){

binary[index]=number%2;

number=number/2;

index++;

}

printf("binary:");

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

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

}

}

else{

printf("not a valid choice:");

}

return 0;

Output: 1.binary to decimal

2. decimal to binary

enter a choice:2

enter a decimal:42

binary:101010

Code to print pattern:-

Input:

#include <stdio.h>

int main() {

int n;

printf("enter n:");

scanf("%d",&n);

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

for(int j=0;j<=i;j++){

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

printf("1");

}

else{

printf("0");

}

}

printf("\n");

} return 0;

}

Output: enter n:5

1

01

101

0101

10101

Code to print pattern:-

Input: #include <stdio.h>

int main() {

int rows = 5;

int i, j;

for (i = 0; i < rows; i++) {

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

printf("%d", j % 2);

}

for (j = 0; j < (rows - i - 1) \* 2; j++) {

printf(" ");

}

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

printf("%d", j % 2);

}

printf("\n");

}

return 0;

}

Output:

0 0

01 01

010 010

0101 0101

0101001010

Code to print Pascal’s triangle:-

Input:

#include <stdio.h>

int main() {

int n;

printf("enter n:");

scanf("%d",&n);

for(int line=0;line<n;line++){

int number=1;

for(int spaces=0;spaces<n-line-1;spaces++){

printf(" ");

}

for(int i=0;i<=line;i++){

printf("%d ",number);

number=number\*(line-i)/(i+1);

}

printf("\n");

}

return 0;

}

Output: enter n:5

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

s