**FUNDAMENTALS OF COMPUTING  
LAB EXPERIMENTS**

**NName J M Prasanth kumar**

**Reg.No:** 192111686

**C-Program**

1. Generation of number series 1, 2, 3, 4,…..n

#include<stdio.h>

#include<conio.h>

int main()

{

int N, i;

printf("Enter the value of N (limit): ");

scanf("%d", &N);

printf("\n");

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

{

if(i==N)

printf("%d", i);

else

printf("%d,", i);

}

getch();

return 0;

}

**OUTPUT:**

1. Generation of even number series 2, 4, 6, …..n

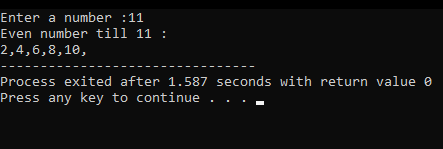
#include<stdio.h>

int main()

{

int n,i;

printf("Enter a number :");

scanf("%d",&n);

printf("Even number till %d :\n",n);

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

{

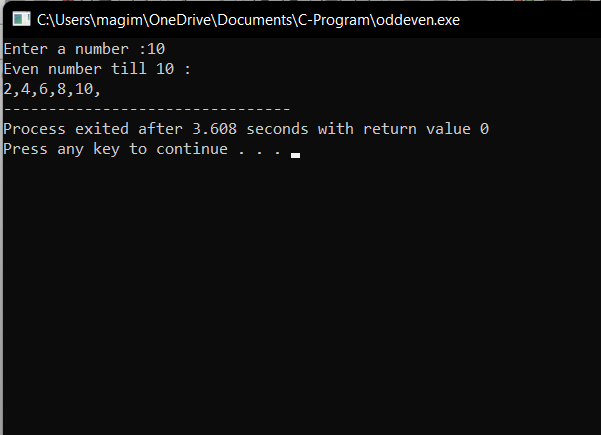
printf("%d,",i);

}

return 0;

}

**OUTPUT:**



1. Generation of ODD number series 1, 3, 5, …..n

#include<stdio.h>

int main()

{

int n,i;

printf("Enter a number :");

scanf("%d",&n);

printf("ODD number till %d :\n",n);

for(i=1;i<=n;i=i+2)

{

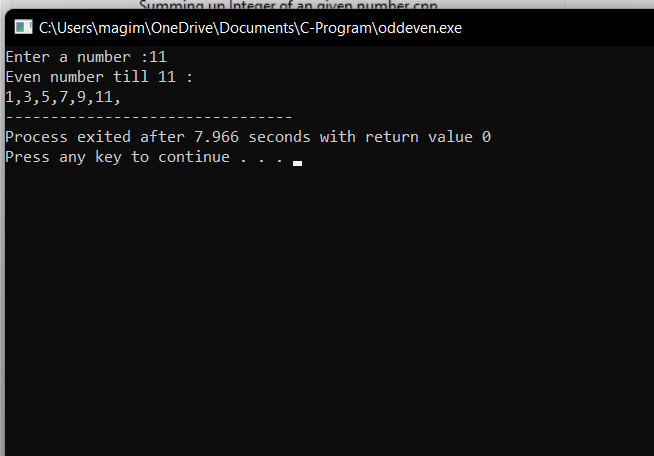
printf("%d,",i);

}

return 0;

}

**OUTPUT:**



1. Generation of Fibonacci series 0, 1, 1, 2, 3, 5, 8, …..n

#include<stdio.h>

int main()

{

int n1=0,n2=1,n3,i,number;

printf("Enter the number of elements:");

scanf("%d",&number);

printf("\n%d %d",n1,n2);

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

{

n3=n1+n2;

printf(" %d",n3);

n1=n2;

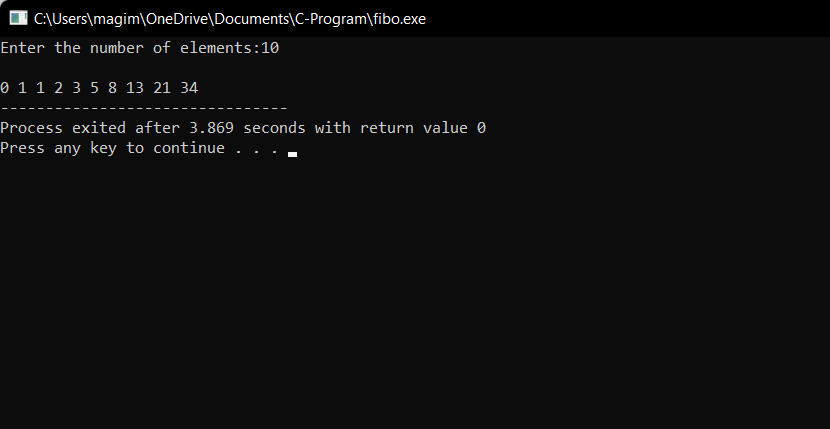
n2=n3;

}

return 0;

}

**OUTPUT:**



1. Summing up series 1 + 2 + 3 + 4….. +n

#include<stdio.h>

int main()

{

int n,i;

int sum=0;

printf("Enter The maximun value of series:");

scanf("%d",&n);

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

printf("Sum Of The Series: ");

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

{

if(i!=n)

{

printf("%d + ",i);

}

else

{

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

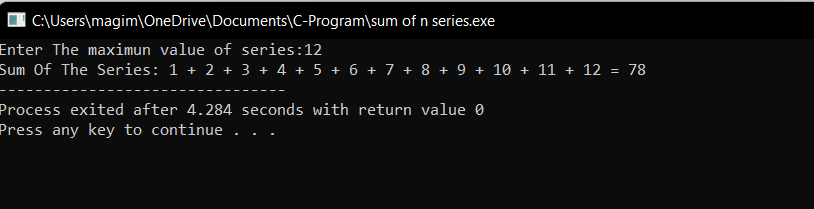
}

}

return 0;

}

**OUTPUT:**



1. Summing up Even Number series

#include <stdio.h>

int main()

{

int i, n, sum=0;

printf("Enter any number: ");

scanf("%d", &n);

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

{

sum += i;

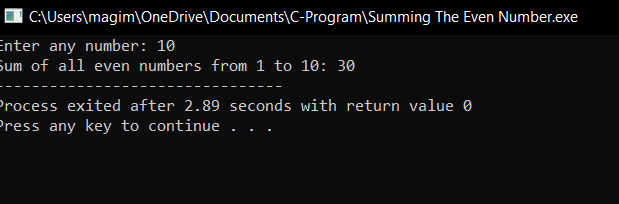
}

printf("Sum of all even numbers from 1 to %d: %d", n, sum);

return 0;

}

**OUTPUT:**



1. Summing up cubes of n numbers

#include<stdio.h>

int main()

{

int n,sum;

printf("enter the value of n:");

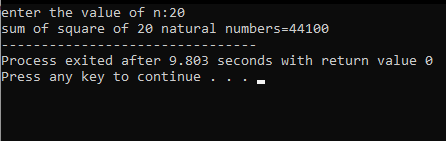
scanf("%d",&n);

sum=(n\*n)((n+1)(n+1))/4;

printf("sum of square of %d natural numbers=%d",n,sum);

}

**OUTPUT:**



1. Finding whether the given integer is odd or even

#include<stdio.h>

int main()

{

int n;

printf("Enter The Number To Check Whether The Given NNumber Is Odd Or Even: ");

scanf("%d",&n);

if(n%2==0)

{

printf("%d Is A Even Number",n);

}

else

{

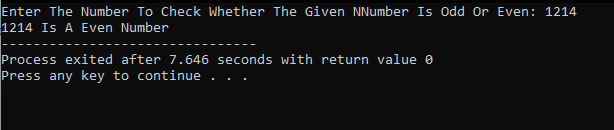
printf("%d Is A Odd Number",n);

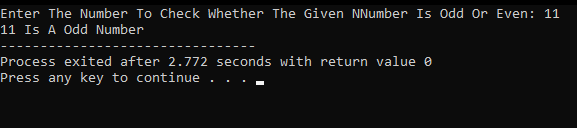
}

return 0;

}

**OUTPUT:**





1. Product series (Factorial of a given number)

#include<stdio.h>

int main()

{

int i,fact=1,number;

printf("Enter a number: ");

scanf("%d",&number);{

if(number<0){

printf("Please enter a positive integer...");

return 0;}

else

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

fact=fact\*i; }

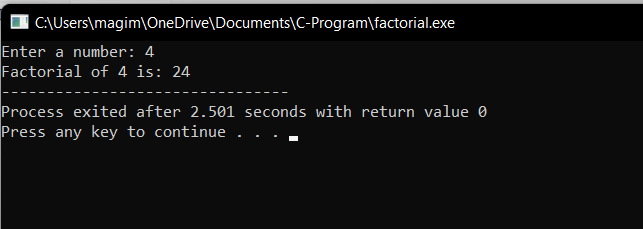
}

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

return 0;

}

**OUTPUT:**



1. Finding given number is Armstrong or not

#include<stdio.h>

int main()

{

int number, temp, rem, sum=0;

printf("Enter number: ");

scanf("%d",&number);

temp=number;

while( number!=0 )

{

rem = number%10;

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

number /= 10;

}

if(temp == sum)

printf("%d is an Armstrong number of order 3.\n",temp);

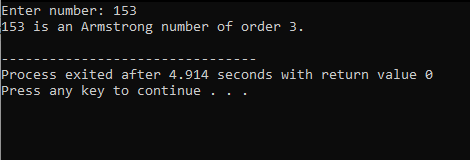
else

printf("%d is not an Armstrong number of order 3.\n",temp);

return 0;

}

**OUTPUT:**



1. Summing up any n numbers and finding average

#include<stdio.h>

int main()

{

int i,n,Sum=0,numbers;

float Average;

printf("\nPlease Enter How many Number you want?\n");

scanf("%d",&n);

printf("\nPlease Enter the elements one by one\n");

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

{

scanf("%d",&numbers);

Sum = Sum +numbers;

}

Average = Sum/n;

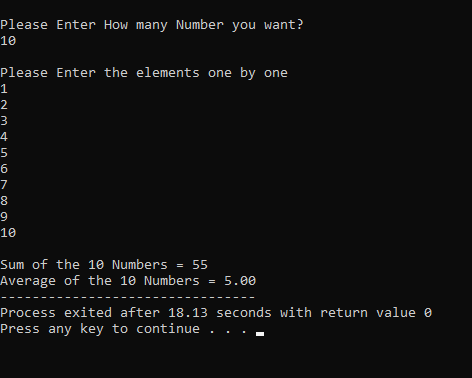
printf("\nSum of the %d Numbers = %d",n, Sum);

printf("\nAverage of the %d Numbers = %.2f",n, Average);

return 0;

}

**OUTPUT:**



1. Printing digits of an integer number

#include <stdio.h>

#define MAX 100

void printDigit(int N)

{

int arr[MAX];

int i = 0,j,r;

while (N != 0) {

r = N % 10;

arr[i] = r;

i++;

N = N / 10;

}

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

printf("%d \n", arr[j]);

}

}

int main()

{

int N;

printf("Enter The Number: ");

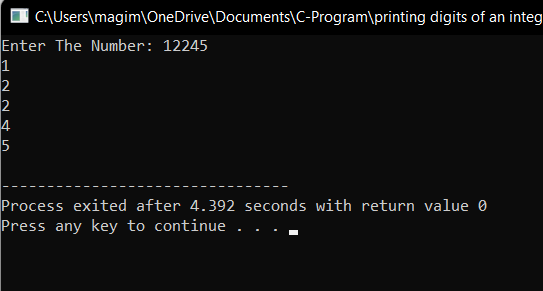
scanf("%d",&N);

printDigit(N);

return 0;

}

**OUTPUT:**



1. Summing up the digits of an integer number

#include <stdio.h>

int getSum(int n)

{

int sum = 0;

while (n != 0) {

sum = sum + n % 10;

n = n / 10;

}

return sum;

}

int main()

{

int n;

printf("Enter The Integer To Sum Up: ");

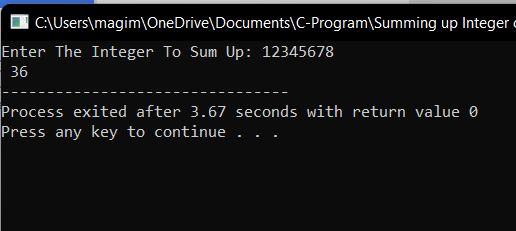
scanf("%d",&n);

printf(" %d ", getSum(n));

return 0;

}

**OUTPUT:**



1. Reversing the digits of an integer number

#include<stdio.h>

int main()

{

int n, rev=0, rem;

printf("Enter a number: ");

scanf("%d", &n);

while(n!=0)

{

rem=n%10;

rev=rev\*10+rem;

n/=10;

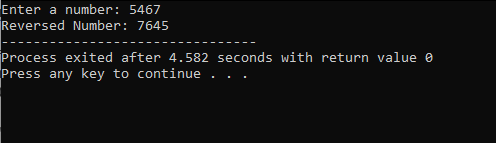
}

printf("Reversed Number: %d",rev);

return 0;

}

**OUTPUT:**



1. Finding the given integer is positive or negative

#include<stdio.h>

int main()

{

int a;

printf("Enter An Integer: ");

scanf("%d",&a);

if(a>0)

{

printf("%d Is A Positive Number...",a);

}

else

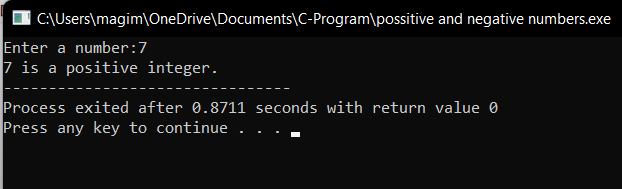
{

printf("%d Is A Negative Number",a);

}

}

**OUTPUT:**



1. Swapping two numbers with a temporary variable

#include<stdio.h>

int main()

{

double first, second, temp;

printf("Enter first number: ");

scanf("%lf", &first);

printf("Enter second number: ");

scanf("%lf", &second);

temp = first;

first = second;

second = temp;

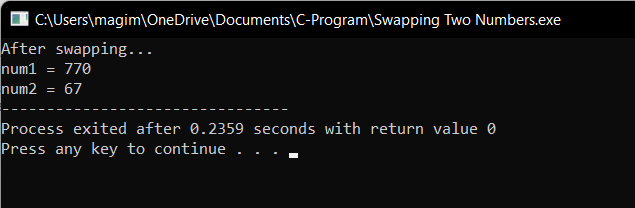
printf("\nAfter swapping, first number = %.2lf\n", first);

printf("After swapping, second number = %.2lf", second);

return 0;

}

**OUTPUT:**



1. Program to convert decimal to hexadecimal `

#include<stdio.h>

int main()

{

long int decimalNumber,remainder,quotient;

int i=1,j,temp;

char hexadecimalNumber[100];

printf("Enter any decimal number: ");

scanf("%ld",&decimalNumber);

quotient = decimalNumber;

while(quotient!=0) {

temp = quotient % 16;

if( temp < 10)

temp =temp + 48; else

temp = temp + 55;

hexadecimalNumber[i++]= temp;

quotient = quotient / 16;

}

printf("Equivalent hexadecimal value of decimal number %d: ",decimalNumber);

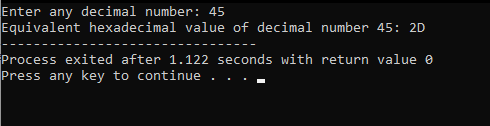
for (j = i -1 ;j> 0;j--)

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

return 0;

}

**OUTPUT:**



1. Program to convet Hexa to decimal

#include <stdio.h>

#include <math.h>

#include <string.h>

int main()

{

char hex[17];

long long decimal, place;

int i = 0, val, len;

decimal = 0;

place = 1;

printf("Enter any hexadecimal number: ");

gets(hex);

len = strlen(hex);

len--;

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

{

if(hex[i]>='0' && hex[i]<='9')

{

val = hex[i] - 48;

}

else if(hex[i]>='a' && hex[i]<='f')

{

val = hex[i] - 97 + 10;

}

else if(hex[i]>='A' && hex[i]<='F')

{

val = hex[i] - 65 + 10;

}

decimal += val \* pow(16, len);

len--;

}

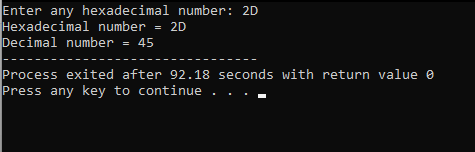
printf("Hexadecimal number = %s\n", hex);

printf("Decimal number = %lld", decimal);

return 0;

}

**OUTPUT:**



1. Program to convert decimal to octal

#include <stdio.h>

int main()

{

int on[10], number, i, j;

printf("\n Please Enter the Number You want to Convert : ");

scanf("%d", &number);

for(i = 0; number > 0; i++)

{

on[i] = number % 8;

number = number / 8;

}

printf("\n Octal Number of a Given Number = ");

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

{

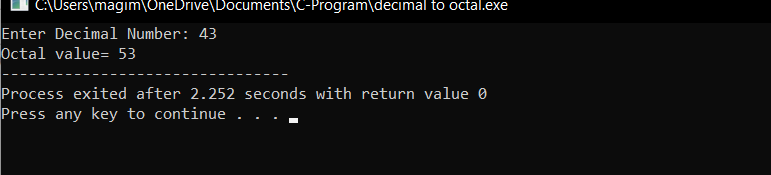
printf("%d", on[j]);

}

return 0;

}

**OUTPUT:**



1. Program to convert octal to decimal

#include <stdio.h>

#include <math.h>

long long od(int on);

int main()

{

int on;

printf("Enter an octal number: ");

scanf("%d", &on);

printf("%d in octal = %lld in decimal", on, od(on));

return 0;

}

long long od(int on)

{

int dn = 0, i = 0;

while(on != 0)

{

dn += (on%10) \* pow(8,i);

++i;

on/=10;

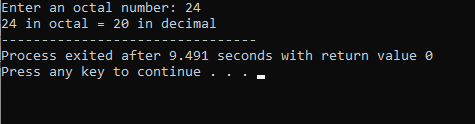
}

i = 1;

return dn;

}

**OUTPUT:**



1. Program to convert binary to decimal

#include <stdio.h>

#include <math.h>

int convert(long long);

int main() {

long long n;

printf("Enter a binary number: ");

scanf("%lld", &n);

printf("%lld in binary = %d in decimal", n, convert(n));

return 0;

}

int convert(long long n)

{

int dec = 0, i = 0, rem;

while (n!=0)

S{

rem = n % 10;

n /= 10;

dec += rem \* pow(2, i);

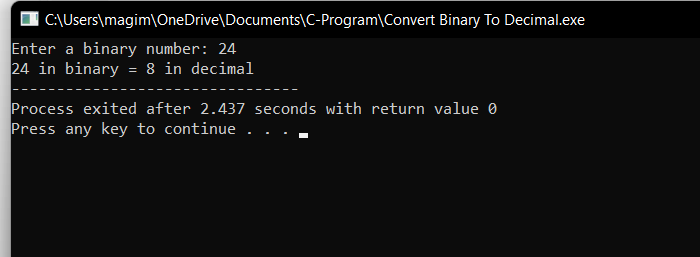
++i;

}

return dec;

}

**OUTPUT:**



1. Write a program for binary addition

#include<stdio.h>

int main()

{

long binary1,binary2;

int i=0,remainder=0,sum[20];

printf("Enter the first binary number:");

scanf("%ld",&binary1);

printf("Enter the second binary number:");

scanf("%ld",&binary2);

while(binary1 !=0||binary2 !=0)

{

sum[i++]=(binary1%10 + binary2%10 + remainder)%2;

remainder=(binary1%10 + binary2%10 + remainder)/2;

binary1=binary1/10;

binary2=binary2/10;

}

if(remainder!=0)

sum[i++]=remainder;

--i;

printf("sum of two binary numbers:");

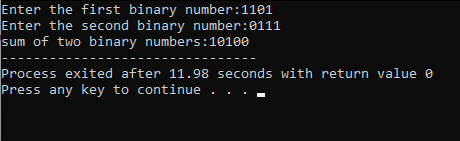
while(i>=0)

printf("%d",sum[i--]);

return 0;

}

**OUTPUT:**



1. Write a program for binary subtraction

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

int binAddition(int a,int b)

{

int c;

while (b != 0)

{

c = (a & b) << 1;

a=a^b;

b=c;

}

return a;

}

int binSubtracton(int a, int b)

{

int carry;

b = binAddition(~b, 1);

while (b != 0)

{

carry = (a & b) << 1;

a = a ^ b;

b = carry;

}

return a;

}

int main()

{

int number1,number2, binAdd, binSub;

printf("Input first Binary value: ");

scanf("%d",&number1);

printf("Input second Binary value: ");

scanf("%d",&number2);

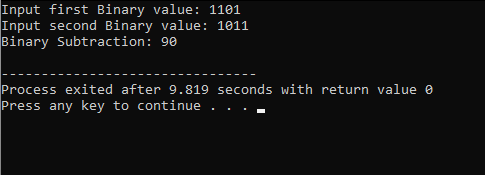
binSub = binSubtracton(number1,number2);

printf("Binary Subtraction: %d\n",binSub);

return 0;

}

**OUTPUT:**



1. Write a program for binary multiplication

#include <stdio.h>

int binaryproduct(int, int);

int main()

{

long binary1, binary2, multiply = 0;

int digit, factor = 1;

printf("Enter the first binary number: ");

scanf("%ld", &binary1);

printf("Enter the second binary number: ");

scanf("%ld", &binary2);

while (binary2 != 0)

{

digit = binary2 % 10;

if (digit == 1)

{

binary1 = binary1 \* factor;

multiply = binaryproduct(binary1, multiply);

}

else

binary1 = binary1 \* factor;

binary2 = binary2 / 10;

factor = 10;

}

printf("Product of two binary numbers: %ld", multiply);

return 0;

}

int binaryproduct(int binary1, int binary2)

{

int i = 0, remainder = 0, sum[20];

int binaryprod = 0;

while (binary1 != 0 || binary2 != 0)

{

sum[i++] =(binary1 % 10 + binary2 % 10 + remainder) % 2;

remainder =(binary1 % 10 + binary2 % 10 + remainder) / 2;

binary1 = binary1 / 10;

binary2 = binary2 / 10;

}

if (remainder != 0)

sum[i++] = remainder;

--i;

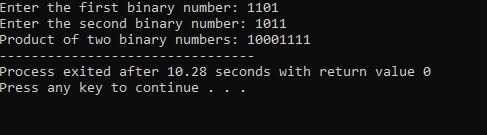
while (i >= 0)

binaryprod = binaryprod \* 10 + sum[i--];

return binaryprod;

}

**OUTPUT:**



**Shell Program**

1. Write a shell script to ask your name, program name and enrollment number and print it on the screen.

Input:

Echo “Enter your name:”  
Read Name  
Echo “Enter your program name:”  
Read Prog  
Echo “Enter your enrollment number:”  
Read Enroll  
Clear  
Echo “Details you entered”  
Echo Name: $Name  
Echo Program Name: $Prog  
Echo Enrolment Number: $Enroll

OutPut:

1. Write a shell script to find the sum, the average and the product of the four integers entered

Input

Echo “Enter four integers with space between”  
Read a b c d  
Sum =`expr $a + $b + $c + $d`  
Avg =`expr $sum / 4`  
Dec =`expr $sum % 4`  
Dec =`expr \ ($dec \\* 1000 \) / 4`  
Product =`expr $a \\* $b \\* $c \\* $d`  
Echo Sum = $sum  
Echo Average = $avg. $dec  
Echo Product = $product

1. Write a shell program to exchange the values of two variables

Input

Echo “Enter value for a:”  
Read a  
Echo “Enter value for b:”  
Read b  
Clear  
Echo “Values of variables before swapping”  
Echo A = $a  
Echo B = $b  
Echo Values of variables after swapping  
a = `expr $a + $b`  
b = `expr $a – $b`  
a = `expr $a – $b`  
Echo A = $a  
Echo B = $b

1. Write a shell script to display the digits which are in odd position in a given 5 digit number

Input

Echo “Enter a 5 digit number”  
Read num  
n = 1  
while [ $n -le 5 ]  
do  
a = `Echo $num | cut -c $n`  
Echo $a  
n = `expr $n + 2`  
done

1. Write a shell program to reverse the digits of five digit integer

Input

Echo “Enter a 5 digit number”  
Read num  
n = $num  
rev=0  
while [ $num -ne 0 ]  
do  
r = `expr $num % 10`  
rev = `expr $rev \\* 10 + $r`  
num = `expr $num / 10`  
done  
Echo “Reverse of $n is $rev”

1. Write a shell program to concatenate two strings and find the length of the resultant string

Input

Echo “Enter first string:”  
Read s1  
Echo “Enter second string:”  
Read s2  
s3 = $s1$s2  
len = `Echo $s3 | wc -c`  
len = `expr $len – 1`  
Echo “Concatenated string is $s3 of length $len ”

1. Write a shell program to find the position of substring in given string

Input

Echo “Enter main string:”  
Read main  
l1 = `Echo $main | wc -c`  
l1 = `expr $l1 – 1`  
Echo “Enter sub string:”  
Read sub  
l2 = `Echo $sub | wc -c`  
l2 = `expr $l2 – 1`  
n = 1  
m = 1  
pos = 0  
while [ $n -le $l1 ]  
do  
a = `Echo $main | cut -c $n`  
b = `Echo $sub | cut -c $m`  
if [ $a = $b ]  
then  
n = `expr $n + 1`  
m = `expr $m + 1`  
pos = `expr $n – $l2`  
r = `expr $m – 1`  
if [ $r -eq $l2 ]  
then  
break  
fi  
else  
pos = 0  
m = 1  
n = `expr $n + 1`  
fi  
done  
Echo “Position of sub string in main string is $pos”

1. Write a shell program to find the gcd for the 2 given numbers

Input

Echo “Enter two numbers with space in between”  
Read a b  
m = $a  
if [ $b -lt $m ]  
then  
m = $b  
fi  
while [ $m -ne 0 ]  
do  
x = `expr $a % $m`  
y = `expr $b % $m`  
if [ $x -eq 0 -a $y -eq 0 ]  
then  
Echo “gcd of $a and $b is $m”  
break  
fi  
m = `expr $m – 1`  
done

1. Write a shell program to add, subtract and multiply the 2 given numbers passed as command line arguments

Input

add = `expr $1 + $2`  
sub = `expr $1 – $2`  
mul = `expr $1 \\* $2`  
Echo “Addition of $1 and $2 is $add”  
Echo “Subtraction of $2 from $1 is $sub”  
Echo “Multiplication of $1 and $2 is $mul”

1. Write a shell script to find the sum, the average and the product of the four integers entered

Input

Echo “Enter four integers with space between”  
Read a b c d  
Sum =`expr $a + $b + $c + $d`  
Avg =`expr $sum / 4`  
Dec =`expr $sum % 4`  
Dec =`expr \ ($dec \\* 1000 \) / 4`  
Product =`expr $a \\* $b \\* $c \\* $d`  
Echo Sum = $sum  
Echo Average = $avg. $dec  
Echo Product = $product

1. Write a shell program to exchange the values of two variables

Input

Echo “Enter value for a:”  
Read a  
Echo “Enter value for b:”  
Read b  
Clear  
Echo “Values of variables before swapping”  
Echo A = $a  
Echo B = $b  
Echo Values of variables after swapping  
a = `expr $a + $b`  
b = `expr $a – $b`  
a = `expr $a – $b`  
Echo A = $a  
Echo B = $b

1. Write a shell program to reverse the digits of five digit integer

Input

Echo “Enter a 5 digit number”  
Read num  
n = $num  
rev=0  
while [ $num -ne 0 ]  
do  
r = `expr $num % 10`  
rev = `expr $rev \\* 10 + $r`  
num = `expr $num / 10`  
done  
Echo “Reverse of $n is $rev”

1. Shell Program to print n natural number

Input

#! /bin/bash  
  
echo "enter the value of n:"  
read n  
for((i=1;i<=n;i++))  
do  
echo " $i"  
  
done

1. Shell Program to find area of different shapes

Input

echo “Enter the side of the square:”

read s

echo “Enter the length and breadth of rectangle:”

read leng

read brea

echo “Enter the radius of the circle:”

read radius

echo “Area of the square is:` expr $s \\* $s ` ”

echo “Area of the rectangle is: `expr $leng \\* $brea`”

echo “Area of the circle is: `expr 3.14 \\* radius \\* radius`”

1. Shell Program to check number is palindrome

Input

#!/bin/bash

# GNU bash Script

n=12321

rev=$(echo $n | rev)

if [ $n -eq $rev ]; then

echo "Number is palindrome"

else

echo "Number is not palindrome"

Fi

1. Shell Program for decimal to binary conversion

Input

echo "Enter the num"

read n

val=0

power=1

while [ $n -ne 0 ]

do

r=`expr $n % 2`

val=`expr $r \\* $power + $val`

power=`expr $power \\* 10`

n=`expr $n \/ 2`

done

echo "Binary equivalent=$val"

1. Shell Program factorial using recursion

Input

echo "Enter a number"

# Read the number

read num

fact=1

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

{

fact=$((fact \* i))

}

echo $fact