***Function Type 3:***

Q) Convert Ferenhit into Celcius

*#include*<stdio.h>

void ferenhitToCelcius(float);

int main(){

    float F;

    printf("Enter temperature Value in ferenhit\n");

    scanf("%f", *&*F);

    ferenhitToCelcius(F);

*return* 0;

}

void ferenhitToCelcius(float F){

*// return ((F-32) \* 5/9);*

    printf("%0.2f Ferenhit = %.2f degree celsius\n",F, ((F*-*32) *\** 5*/*9));

*// f-32 \*\* 5/9*

}

Q)Find Area And perimeter of rectangle and circle

*#include*<stdio.h>

void findAreaNPerimeterOfRect(int, int);

void findAreaNPerimeterOfCircle(float);

void findAreaNPerimeterOfRect(int length, int breadth){

    printf("Area of rectangle is %d\n", length*\**breadth);

    printf("perimeter of rectangle is %d\n", 2 *\** (length*+*breadth));

*// return length\*breadth;*

}

void findAreaNPerimeterOfCircle(float radius){

*// return 2 \* 3.14 \* radius;*

    printf("Area of circle is %.2f\n", 3.14 *\** radius *\** radius);

    printf("perimeter of circle is %.2f\n", 2 *\** 3.14 *\** radius);

}

int main(){

    int length, breadth;

    printf("Enter Length and breadth for finding Area of rectangle\n");

    scanf("%d %d", *&*length, *&*breadth);

    findAreaNPerimeterOfRect(length, breadth);

    float radius;

    printf("Enter radius value for finding Perimeter of circle\n");

    scanf("%f", *&*radius);

    findAreaNPerimeterOfCircle(radius);

*return* 0;

}

Q)Find sum of digits of number and reverse

*// find sum of 3 digit num and reverse it*

*#include*<stdio.h>

void findSumOfDigits(int);

void findReverse(int, int,int,int);

int main(){

    int num;

    printf("Enter a 3 digit number to find sum of digit: \n");

    scanf("%d", *&*num);

    findSumOfDigits(num);

*return* 0;

}

void findSumOfDigits(int num){

    int temp *=* num;

    int r1 *=* num *%*10; *//3*

    num *=* num */*10; *//12*

    int r2 *=* num *%*10;

    int r3 *=* num */*10;

    printf("Sum of %d is %d\n", temp, r1*+*r2*+*r3);

    findReverse(r1,r2,r3, num);

*// return r1+r2+r3;*

*// printf("Reverse num of %d is %d\n", temp, (r1\*100)+(r2\*10)+r3);*

}

void findReverse(int r1, int r2, int r3, int num){

*// printf("Sum of %d is %d\n", temp, r1+r2+r3);*

*// return (r1\*100)+(r2\*10)+r3;*

    printf("Reverse num of %d is %d\n", num, (r1*\**100)*+*(r2*\**10)*+*r3);

}

Q)Find Even or Odd

*#include* <stdio.h>

void checkEvenOdd(int);

int main()

{

    int num;

    printf("Enter Number:\n");

    scanf("%d", *&*num);

*// checkEvenOdd() ? printf("Even Number\n") : printf("Odd Number\n");*

    checkEvenOdd(num);

*return* 0;

}

void checkEvenOdd(int num)

{

    num *%* 2 *?* printf("%d is odd num\n", num) *:* printf("%d is even num\n", num);

*// return !num%2;2*

}

Q)Find Salary after calculating da,ta, hra

*#include*<stdio.h>

*// if basic <= 5000 da,ta, hra -> 10%,20,30*

*// otherwise 15,25,30*

void calSalary(float);

int main(){

    float basic;

    printf("Enter Basic of Salary\n");

    scanf("%f", *&*basic);

*// printf("Salaray is %.2f\n",calSalary());*

    calSalary(basic);

    printf("End\n");

*return* 0;

}

void calSalary(float basic){

    float salary;

*if*(basic *<=*5000){

        salary *=* basic *+* (basic *\** 10)*/*100 *+* (basic *\** 20)*/*100 *+* (basic*\**30)*/*100;

    }

*else*

        salary *=* basic *+* (basic *\** 15)*/*100 *+* (basic *\** 25)*/*100 *+* (basic*\**30)*/*100;

    printf("Salary is %.2f\n", salary);

*// return salary;*

}

Q) Swap two Variables

*#include*<stdio.h>

void swapToNum(int, int);

int main(){

    int a,b;

    printf("Enter the value of a:\n");

    scanf("%d", *&*a);

    printf("Enter the value of b:\n");

    scanf("%d", *&*b);

    swapToNum(a,b);

    printf("End\n");

*return* 0;

}

void swapToNum(int a, int b){

*// ---before swapping----*

    printf("---before swapping----\n");

    printf("a=%d, b=%d\n",a,b);

*//---after swapping----*

    int temp *=* a;

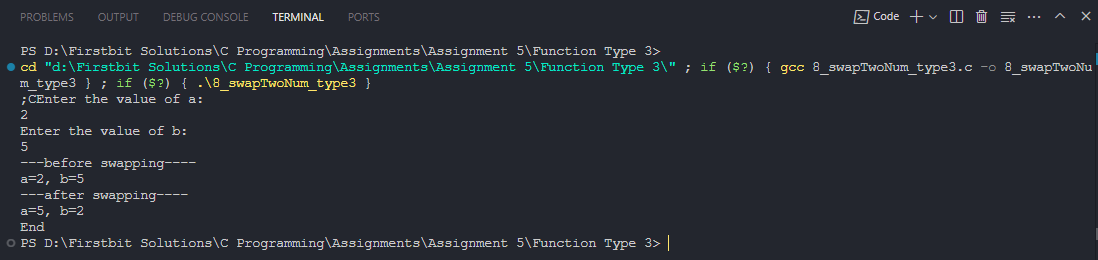
    a *=* b;

    b *=* temp;

    printf("---after swapping----\n");

    printf("a=%d, b=%d\n",a,b);

}



Q) Find driver eligibility

*#include*<stdio.h>

*//age must be greater than 40*

*//d exp > 10*

*//12th marks > 60*

void checkDriverEligibility(int, int, int);

int main(){

    int age,exp,marks;

    printf("Enter Age: ");

    scanf("%d", *&*age);

    printf("Enter Exp: ");

    scanf("%d", *&*exp);

    printf("Enter Marks: ");

    scanf("%d", *&*marks);

    checkDriverEligibility(age, exp, marks);

*// checkDriverEligibility() ? printf("Driver is Eligible\n") : printf("Driver is Not Eligible\n");*

    printf("End\n");

*return* 0;

}

void checkDriverEligibility(int age, int exp, int marks){

    (age *>*40 *&&* exp *>* 10 *&&* marks *>* 60) *?* printf("Driver is eligible") *:*printf("Driver is NOT eligible");

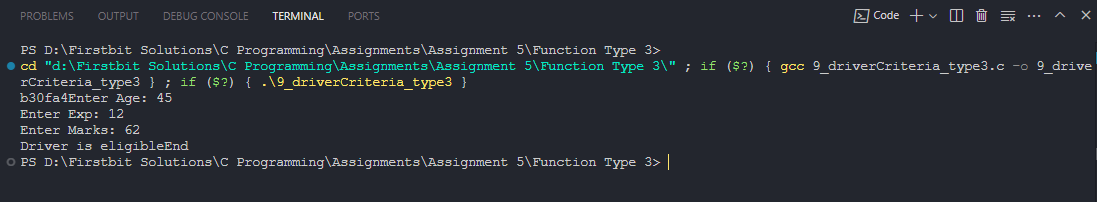
*// if(age >40 && exp > 10 && marks > 60){*

*//     return 1;*

*// }*

*// return 0;*

}



Q)Discount on price

*#include*<stdio.h>

void calNetPrice(int, int);

int main(){

    int price, discount;

    printf("Enter price:\n");

    scanf("%d", *&*price);

    printf("Enter discount percentage Example: 20\n");

    scanf("%d", *&*discount);

*// printf("Your net price to be paid is:  %d\n", calNetPrice());*

    calNetPrice(price, discount);

    printf("End\n");

*return* 0;

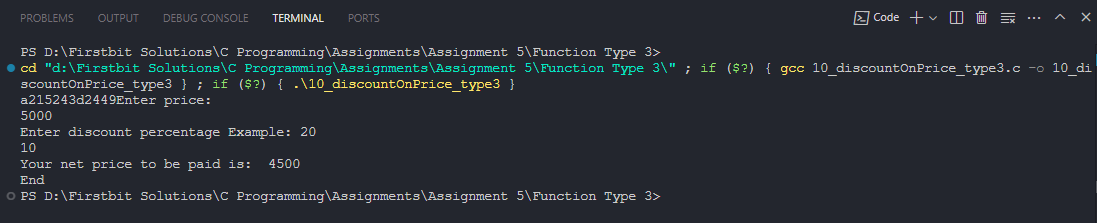
}

void calNetPrice(int price, int discount){

    printf("Your net price to be paid is:  %d\n", price *-* ((price*\**discount)*/*100));

*// return price - (price\*discount)/100;*

}



Q) Find Greatest of 3 using nested if

*#include*<stdio.h>

void findGreatest(int, int, int);

int main(){

    printf("Start\n");

    int a,b,c;

    printf("Enter the value of a: ");

    scanf("%d",*&*a);

    printf("Enter the value of b: ");

    scanf("%d",*&*b);

    printf("Enter the value of c: ");

    scanf("%d",*&*c);

    findGreatest(a,b,c);

*// printf("%d is greatest\n",findGreatest(a,b,c));*

    printf("End\n");

*return* 0;

}

void findGreatest(int a, int b,int c){

    int greatest;

*if*(a*>*b){

*if* (a*>*c)

        {

*// printf("A is Greatest of three.\n");*

            greatest *=* a;

*// return a;*

        }

*else*

        {

*// printf("C is Greatest of three.\n");*

            greatest *=* c;

*// return c;*

        }

    } *else* {

*if*(b*>*c){

*// printf("B is Greatest of three.\n");*

*// return b;*

            greatest *=* b;

        }

*else* {

*// printf("C is Greatest of three.\n");*

*// return c;*

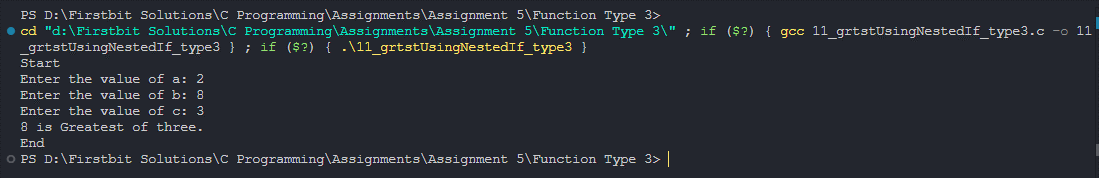
            greatest *=* c;

        }

    }

    printf("%d is Greatest of three.\n", greatest);

}



Q) Accept two numbers from user and an operator (+,-,/,\*,%) based on that perform the desired operations.

*#include* <stdio.h>

*// for type 2, there ain't return type, so we can not send any value to function for process, so again we have to take input again and again, doesn't make sense, it'll increase repeatation*

void showChoices(int, int ,char);

int main()

{

    printf("Start\n");

    int num1, num2;

    char operator;

    printf("Enter the value of number 1: ");

    scanf("%d", *&*num1);

    printf("Enter the value of number 2: ");

    scanf("%d", *&*num2);

    printf("------Enter your choice :------\n");

    printf("For Addition enter '+'\n");

    printf("For Subtraction enter '-'\n");

    printf("For Multiplication enter '\*'\n");

    printf("For Divivsion enter '/'\n");

    printf("For Modulo enter '*%*'\n");

    fflush(stdin);

    scanf("%c", *&*operator);

*// printf("Answer is %d\n", showChoices());*

    showChoices(num1, num2, operator);

    printf("End\n");

*return* 0;

}

void showChoices(int num1, int num2, char operator){

*if* (operator*==* '+')

    {

        printf("%d %c %d = %d\n", num1, operator, num2, num1 *+* num2);

*// return num1+num2;*

    }

*else* *if* (operator*==* '-')

    {

        printf("%d %c %d = %d\n", num1, operator, num2, num1 *-* num2);

*// return num1-num2;*

    }

*else* *if* (operator*==* '\*')

    {

        printf("%d %c %d = %d\n", num1, operator, num2, num1 *\** num2);

*// return num1\*num2;*

    }

*else* *if* (operator*==* '/')

    {

        printf("%d %c %d = %d\n", num1, operator, num2, num1 */* num2);

*// return num1 / num2;*

    }

*else* *if* (operator*==* '%')

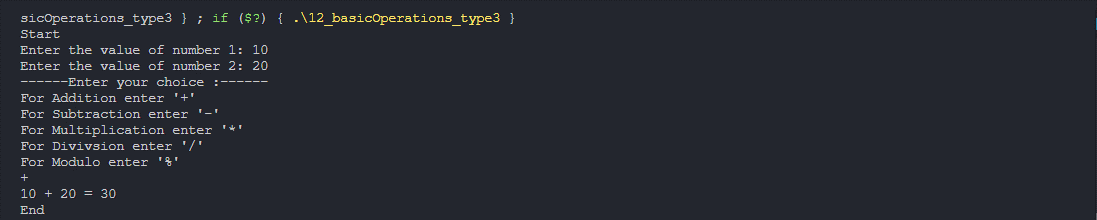
    {

        printf("%d %c %d = %d\n", num1, operator, num2, num1 *%* num2);

*// return num1 % num2;*

    }

}



Q) Display a menu to the user (like 1.Even Odd 2. Basic salary etc), ask the user to enter his choice, then based on that perform the desired operations.

*#include* <stdio.h>

void checkEvenOdd(int);

void calculateSalary(float);

int main()

{

    int choiceNum;

    printf("----Choices-----\n");

    printf("Enter 1 for calculate Even odd\n");

    printf("Enter 2 for calculate Salary\n");

    scanf("%d", *&*choiceNum);

*if* (choiceNum *==* 1)

    {

*// checkEvenOdd() ? printf("EVEN number\n") : printf("ODD number\n");*

        int num;

        printf("Enter number to check weather number is even or odd.\n");

        scanf("%d", *&*num);

        checkEvenOdd(num);

    }

*else* *if* (choiceNum *==* 2)

    {

*// printf("Salary is %.2f\n",calculateSalary());*

        float basic;

        printf("Enter Basic salary\n");

        scanf("%f", *&*basic);

        calculateSalary(basic);

    }

*else*

    {

        printf("Invalid Choice\n");

    }

*return* 0;

}

void checkEvenOdd(int num)

{

    num *%* 2 *?* printf("%d is Odd Number\n", num) *:* printf("%d is Even number\n", num);

*// return !num%2;*

}

void calculateSalary(float basic)

{

    float salary;

*if* (basic *<=* 5000)

    {

        salary *=* basic *+* (basic *\** 10) */* 100 *+* (basic *\** 20) */* 100 *+* (basic *\** 30) */* 100;

    }

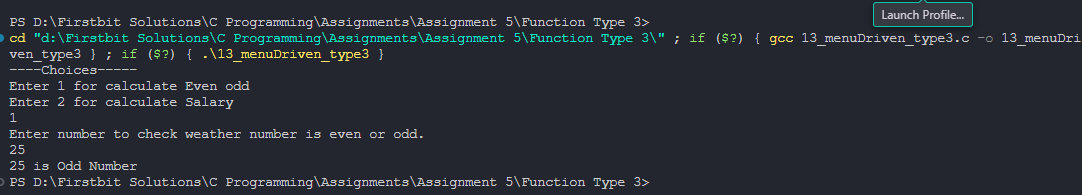
*else*

        salary *=* basic *+* (basic *\** 15) */* 100 *+* (basic *\** 25) */* 100 *+* (basic *\** 30) */* 100;

    printf("Salary is %.2f\n", salary);

*// return salary;*

}



Q) Accept the price from user. Ask the user if he is a student (user may say yes or no). If he is a student and he has purchased more than 500 than discount is 20% otherwise discount is 10%. But if he is not a student then if he has purchased more than 600 discount is 15% otherwise there is not discount.

*#include* <stdio.h>

void checkIsStudent(char, float);

int main()

{

    printf("Start\n");

    float price;

    int discount;

    printf("\nEnter price: ");

    scanf("%f", *&*price);

    char checkStudent;

    printf("If your are student than press 'Y', else press 'N'\n");

    fflush(stdin);

    scanf("%c", *&*checkStudent);

    checkIsStudent(checkStudent, price);

*// printf("You got %d\% Discount\n", discount);*

*// printf("You have to pay %.2f rs.\n", price - (price \* discount) / 100);*

    printf("End\n");

*return* 0;

}

void checkIsStudent(char checkStudent, float price)

{

    int discount;

*if* (checkStudent *==* 'y' *||* checkStudent *==* 'Y')

    {

        discount *=* price *>* 500 *?* 20 *:* 10;

*// return 1;*

    }

*else* *if* (checkStudent *==* 'n' *||* checkStudent *==* 'N')

    {

        discount *=* price *>* 600 *?* 15 *:* 0;

*// return 0;*

    }

*else*

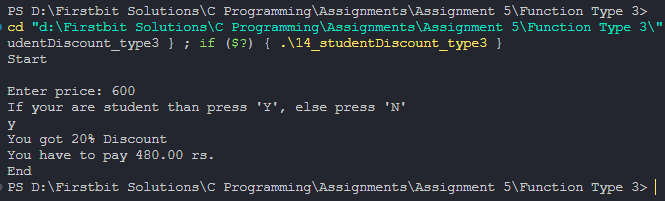
        printf("Invalid choice for student elligibility\n");

    printf("You got %d*\%* Discount\n", discount);

    printf("You have to pay %.2f rs.\n", price *-* (price *\** discount) */* 100);

*// return 0;*

}



Q)Print 1 to 10

*#include*<stdio.h>

*/\**

*--------#########---------*

*NO Change in code at all compare to type 1 function, as there nothing to return and no parameter, we are printing only.*

*--------#########------------*

*\*/*

void print1To10();

int main(){

    printf("Start\n");

    print1To10();

    printf("End\n");

*return* 0;

}

void print1To10(){

*for* (int i *=* 1; i *<* 11; i*++*)

    {

        printf("%d\n", i);

    }

}

Q) Print table for the given number.

*#include* <stdio.h>

void printTable(int);

int main()

{

    printf("Start\n");

    int n;

    printf("Enter Any Number u want to print table of\n");

    scanf("%d", *&*n);

    printTable(n);

    printf("End\n");

*return* 0;

}

void printTable(int n) {

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

    {

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

    }

}

Q)Sum of number in given range

*#include*<stdio.h>

void sumInRange(int, int);

int main(){

    int lower, upper;

    printf("Enter Lower limit and Upper limit of Range\n");

    scanf("%d %d", *&*lower, *&*upper);

    sumInRange(lower, upper);

*return* 0;

}

void sumInRange(int lower, int upper){

    int sum*=*0;

*for* (int i *=* lower; i *<=* upper; i*++*)

    {

        sum *+=* i;

    }

    printf("Sum of Num from %d to %d is %d", lower, upper, sum);

}

Q)Check Prime Number

*#include* <stdio.h>

void checkPrime(int);

int main()

{

    printf("Start\n");

*// checkPrime() ? printf("Num is Prime Number\n"): printf("Num is NOT a Prime Number\n");*

    int num;

    printf("Enter number u want to check prime of\n");

    scanf("%d", *&*num);

    checkPrime(num);

    printf("End\n");

*return* 0;

}

void checkPrime(int num){

    int isPrime *=* 1;

*for* (int i *=* 2; i*\**i *<=* num; i*++*)

    {

*if* (num *%* i *==* 0)

        {

            isPrime *=* 0;

        }

    }

    isPrime *?* printf("Num is Prime Number\n")*:* printf("Num is NOT a Prime Number\n");

}

Q)Check Armstrong Number

*#include* <stdio.h>

*#include* <math.h>

*// if 123 is num, and 1^3 + 2^3 + 3^3 = 123, then its armstrong num*

*//example 153 = 1 + 125+ 27 is armstrong num*

*//1634 = 1^4 + 6^4 + 3^4 + 4^4*

void checkArmStrong(int);

int main()

{

    printf("Start\n");

*// checkArmStrong() ? printf("Number is Armstrong\n") : printf("Number is Not an Armstrong\n");*

    int num;

    printf("Enter a number:\n");

    scanf("%d", *&*num);

    checkArmStrong(num);

    printf("End\n");

*return* 0;

}

void checkArmStrong(int num){

    int temp *=* num, sum *=* 0, count *=*0;

*//find length of number to find exponent*

*while*(temp*>*0){

        count*++*;

        temp */=* 10;

    }

    printf("Count = %d\n", count);

*//temp becomes 0, so ressign for further use*

    temp *=* num;

*while* (temp *>* 0)

    {

        int rem *=* temp *%* 10;

*//cal power of rem*

        int power *=* 1, tempCount *=* count;

*for* (int i *=* 1; i *<=* tempCount; i*++*)

        {

            power *\*=* rem;

        }

*// printf("Power = %d\n", power);*

        sum *+=* power;

        temp */=* 10;

    }

    sum *==* num *?* printf("Number is Armstrong\n") *:* printf("Number is Not an Armstrong\n");

*// return sum == num;*

}

Q)Perfect Number

*#include*<stdio.h>

*//number can be called perfect if, sum of its divisors is same as number itself*

*//ex: 6 because 1 + 2 + 3 = 6*

*//28 beacuse, 1 +2 + 4 + 7 + 14 =28*

void checkPerfectNum(int);

int main(){

    printf("Start\n");

    int num;

    printf("Enter a num:\n");

    scanf("%d", *&*num);

*// checkPerfectNum() ? printf("Num is a Perfect number\n"):printf("Num is NOT a Perfect number\n") ;*

    checkPerfectNum(num);

    printf("End\n");

*return* 0;

}

void checkPerfectNum(int num){

    int temp *=* num, sum *=*0;

*for* (int i *=* 1; i *<=* temp*/*2; i*++*)

    {

*if*(num*%*i*==*0) sum *+=* i;

    }

*// return temp==sum;*

*if*(temp*==*sum) printf("%d is a Perfect number\n", num);

*else* printf("%s is not a Perfect num\n", num);

}

Q)Find Factorial

*#include*<stdio.h>

void findFactorial(int);

int main(){

    printf("Start\n");

    int num;

    printf("Enter a number:\n");

    scanf("%d", *&*num);

*// printf("Answer is %d\n", findFactorial());*

    findFactorial(num);

    printf("End\n");

*return* 0;

}

void findFactorial(int num) {

    int temp *=*num, fact *=*1;

*for*(int i*=*num; i*>*0;i*--*){

        fact *\*=* i;

    }

    printf("%d! = %d\n",temp, fact);

*// return fact;*

}

Q)Strong Number

*#include* <stdio.h>

*// num is called strong if its sum of its digit's factorial is same as num*

*// ex: 145, 1! + 4!+ 5! = 145*

int checkStrongNum();

int main()

{

    printf("Start\n");

    int num;

    printf("Enter a number:\n");

    scanf("%d", *&*num);

*// checkStrongNum() ? printf("Num is Strong Num\n") : printf("Num is NOT a Strong Num\n");*

    checkStrongNum(num);

    printf("End\n");

*return* 0;

}

int checkStrongNum(int num) {

    int temp *=* num, rem, sum *=* 0;

*while* (temp *>* 0)

    {

        rem *=* temp *%* 10;

*//--------Factorial Calculation-------*

*// find factorial of rem*

        int factorial *=* 1;

*while* (rem *>* 0)

        {

            factorial *\*=* rem;

            rem*--*;

        }

*// add factorial of rem to sum*

        sum *+=* factorial;

*// continue*

        temp */=* 10;

    }

*// return sum==num;*

*if* (sum *==* num)

        printf("%d is a Strong Number\n", num);

*else*

        printf("%d is NOT a Strong Number\n", num);

}

Q) Palindrome

*#include* <stdio.h>

*// 121, 1331, 12321*

void checkPalindrome(int);

int main()

{

    printf("Start\n");

    int num;

    printf("Enter a number:\n");

    scanf("%d", *&*num);

    checkPalindrome(num);

    printf("End\n");

*return* 0;

}

void checkPalindrome(int num) {

    int temp *=* num, rem, rev *=* 0;

*while* (temp *>* 0)

    {

        rem *=* temp *%* 10;

        rev *=* rev *\** 10 *+* rem;

        temp */=* 10;

    }

*if* (num *==* rev)

        printf("%d is a Palindrome number\n", num);

*else*

        printf("%d is NOT a Palindrome number\n", num);

}

Q)Add first dig and last dig of a num and show summation

*#include*<stdio.h>

*#include*<math.h>

*//add first and and last digit of given num*

void findFirstNLastDigSum(int);

int main(){

    printf("Start\n");

    int num;

    printf("Enter a number:\n");

    scanf("%d", *&*num);

*// printf("Sum of first and last digit of number is %d\n", findFirstNLastDigSum());*

    findFirstNLastDigSum(num);

    printf("End\n");

*return* 0;

}

void findFirstNLastDigSum(int num){

    int temp *=* num, lastDigit, firstDigit, lengthOfNum*=*0;

    lastDigit *=* temp*%*10;

*//logic 2 for find 1st digit of num*

*while* (temp*>*0)

    {

        firstDigit *=* temp*%*10;

        temp */=* 10;

    }

*// return firstDigit + lastDigit;*

    printf("Sum of first digit(%d) + last Digit(%d) = %d\n", firstDigit, lastDigit, firstDigit*+*lastDigit);

}

Q)Print Armstrong number in range 1 to n

*#include* <stdio.h>

*#include* <math.h>

*// if 123 is num, and 1^3 + 2^3 + 3^3 = 123, then its armstrong num*

*// example 153 = 1 + 125+ 27 is armstrong num*

*// 1634 = 1^4 + 6^4 + 3^4 + 4^4*

void printArmstrongNum();

int main()

{

    printf("Start\n");

    int num;

    printf("Enter a number upto which u want to armstrong nums:\n");

    scanf("%d", *&*num);

*for* (int i *=* 1; i *<=* num; i*++*)

    {

    printArmstrongNum(i);

    }

    printf("End\n");

*return* 0;

}

void printArmstrongNum(int i)

{

        int temp *=* i, sum *=* 0, count *=* 0;

*// find length of number to find exponent*

*while* (temp *>* 0)

        {

            count*++*;

            temp */=* 10;

        }

*// temp becomes 0, so ressign for further use*

        temp *=* i;

*while* (temp *>* 0)

        {

            int rem *=* temp *%* 10;

*// cal power of rem*

            int power *=* 1, exponent *=* count;

*while* (exponent*--*)

            {

                power *\*=* rem;

            }

            sum *+=* power;

            temp */=* 10;

        }

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

}

Q)Check Prime number in Range

*#include* <stdio.h>

void checkPrime(int num)

{

    int isPrime;

*for* (int i *=* 2; i *\** i *<=* num; i*++*)

    {

        isPrime *=* 1;

*if* (num *%* i *==* 0)

        {

            isPrime *=* 0;

*break*;

        }

    }

*if* (isPrime)

    {

        printf("%d ", num);

    }

}

int main()

{

    int num, isPrime *=* 1, j;

    printf("Enter number upto which u want to check prime of\n");

    scanf("%d", *&*num);

*for* (int i *=* 1; i *<=* num; i*++*)

    {

        checkPrime(i);

    }

*return* 0;

}

Q)Check Perfect num in range

*#include* <stdio.h>

*// number can be called perfect if, sum of its divisors is same as number itself*

*// ex: 6 because 1 + 2 + 3 = 6*

*// 28 beacuse, 1 +2 + 4 + 7 + 14 =28*

void checkPerfect(int num)

{

    int sum *=* 0;

*for* (int i *=* 1; i *< num* ; i*++*)

    {

*if*(num*%*i *==*0) sum *+=* i;

    }

*if* (num *==* sum)

        printf("%d ", num);

}

int main()

{

    int num;

    printf("Upto which range u want to check perfect num: ");

    scanf("%d", *&*num);

*for* (int i *=* 1; i *<=* num; i*++*)

    {

        checkPerfect(i);

    }

*return* 0;

}

Q) Check Strong number in range 1 to n

*#include*<stdio.h>

*//num is called strong if its sum of its digit's factorial is same as num*

*//ex: 145, 1! + 4!+ 5! = 145*

*//not optimized as in will open and close function stack frame multiple time*

void checkStrong(int num){

    int temp *=* num, rem, sum*=*0;

*while*(temp*>*0){

        rem *=* temp*%*10;

*//--------Factorial Part-------*

*//find factorial of rem*

        int factorial*=*1;

*while*(rem*>*0){

            factorial *\*=* rem;

            rem*--*;

        }

*//add factorial of rem to sum*

        sum *+=* factorial;

*//continue*

        temp */=* 10;

    }

*if*(sum*==*num) printf("%d ", num);

}

int main(){

    int num;

    printf("Enter a number:\n");

    scanf("%d", *&*num);

*for* (int i *=* 1; i *<=* num; i*++*)

    {

        checkStrong(i);

    }

*return* 0;

}

Q)Print Fibonacci upto n number

*#include*<stdio.h>

*//0 1 1 2 3 5 8 13 21 34 55*

void printFibonacciInRange(int);

int main(){

    printf("Start\n");

    int num;

    printf("Enter a number upto which u want to print fibonacci series\n");

    scanf("%d", *&*num);

    printFibonacciInRange(num);

    printf("End\n");

*return* 0;

}

void printFibonacciInRange(int num) {

    int first *=*0, second *=* 1, next *=* 0;

*while* (next*<=*num)

    {

        printf("%d ", next);

        first *=* second;

        second *=* next;

        next *=* first *+* second;

    }

}