a.

* #include <stdio.h>

int main() {

int year;

printf("Enter a year: ");

scanf("%d", &year);

if (year % 400 == 0) {

printf("%d is a leap year.", year);

}

else if (year % 100 == 0) {

printf("%d is not a leap year.", year);

}

// leap year if not divisible by 100

// but divisible by 4

else if (year % 4 == 0) {

printf("%d is a leap year.", year);

}

// all other years are not leap year

else {

printf("%d is not a leap year.", year);

}

return 0;

}

**Output 1**

**Enter a year: 1900**

**1900 is not a leap year**

* #include <stdio.h>

int main() {

long long n;

int count = 0;

printf("Enter an integer: ");

scanf("%lld", &n);

while (n != 0) {

n /= 10; // n = n/10

++count;

}

printf("Number of digits: %d", count);

}

**Output**

**Enter an integer: 3452**

**Number of digits: 4**

**b.**

* #include <stdio.h>

int main() {

int num, originalNum, remainder, result = 0;

printf("Enter a three-digit integer: ");

scanf("%d", &num);

originalNum = num;

while (originalNum != 0) {

remainder = originalNum % 10;

result += remainder \* remainder \* remainder;

originalNum /= 10;

}

if (result == num)

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

else

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

return 0;

}

**Output**

**Enter a three-digit integer: 371**

**371 is an Armstrong number**

* #include <stdio.h>

**int** main()

{

**float** fh,cl;

**int** choice;

printf("\n1: Convert temperature from Fahrenheit to Celsius.");

printf("\n2: Convert temperature from Celsius to Fahrenheit.");

printf("\nEnter your choice (1, 2): ");

scanf("%d",&choice);

**if**(choice ==1){

printf("\nEnter temperature in Fahrenheit: ");

scanf("%f",&fh);

cl= (fh - 32) / 1.8;

printf("Temperature in Celsius: %.2f",cl);

}

**else** **if**(choice==2){

printf("\nEnter temperature in Celsius: ");

scanf("%f",&cl);

fh= (cl\*1.8)+32;

printf("Temperature in Fahrenheit: %.2f",fh);

}

**else**{

printf("\nInvalid Choice !!!");

}

**return** 0;

}

**Output**

1: Convert temperature from Fahrenheit to Celsius.

2: Convert temperature from Celsius to Fahrenheit.

Enter your choice (1, 2): 1

Enter temperature in Fahrenheit: 98.6

Temperature in Celsius: 37.00

**c.**

* #include <stdio.h>

void checkPrimeAndDisplay(int n);

int main()

{

int n;

printf("Enter a positive integer: ");

scanf("%d",&n);

// n is passed to the function

checkPrimeAndDisplay(n);

return 0;

}

// return type is void meaning doesn't return any value

void checkPrimeAndDisplay(int n)

{

int i, flag = 0;

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

{

if(n%i == 0){

flag = 1;

break;

}

}

if(flag == 1)

printf("%d is not a prime number.",n);

else

printf("%d is a prime number.", n);

}

**Output**

Enter a positive integer: 3

3 is a prime number.

* #include <math.h>

#include <stdio.h>

int main() {

double a, b, c, discriminant, root1, root2, realPart, imagPart;

printf("Enter coefficients a, b and c: ");

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

discriminant = b \* b - 4 \* a \* c;

// condition for real and different roots

if (discriminant > 0) {

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

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

printf("root1 = %.2lf and root2 = %.2lf", root1, root2);

}

// condition for real and equal roots

else if (discriminant == 0) {

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

printf("root1 = root2 = %.2lf;", root1);

}

// if roots are not real

else {

realPart = -b / (2 \* a);

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

printf("root1 = %.2lf+%.2lfi and root2 = %.2f-%.2fi", realPart, imagPart, realPart, imagPart);

}

return 0;

}

**Output**

**Enter coefficients a, b and c: 2.3**

**4**

**5.6**

**root1 = -0.87+1.30i and root2 = -0.87-1.30i**

**d.**

* #include <stdio.h>

int main()

{

int i, num, sum = 0;

/\* Input a number from user \*/

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)

{

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

}

else

{

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

}

return 0;

}

**Output**

**Enter any number to check perfect number: 6**

**6 is perfect number**

#include<stdio.h>

#include<math.h>

int main()

{

printf("Enter a Number to Find Factorial: ");

printf("\nFactorial of a Given Number is: %d ",fact());

return 0;

}

int fact()

{

int i,fact=1,n;

scanf("%d",&n);

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

{

fact=fact\*i;

}

return fact;

}

OUTPUT:

Enter a Number to Find Factorial: 5

Factorial of a Given Number is: 120

**e.**

* #include <stdio.h>

/\* Function declarations \*/

int max(int num1, int num2);

int min(int num1, int num2);

int main()

{

int num1, num2, maximum, minimum;

/\* Input two numbers from user \*/

printf("Enter any two numbers: ");

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

maximum = max(num1, num2); // Call maximum function

minimum = min(num1, num2); // Call minimum function

printf("\nMaximum = %d\n", maximum);

printf("Minimum = %d", minimum);

return 0;

}

/\*\*

\* Find maximum between two numbers.

\*/

int max(int num1, int num2)

{

return (num1 > num2 ) ? num1 : num2;

}

/\*\*

\* Find minimum between two numbers.

\*/

int min(int num1, int num2)

{

return (num1 > num2 ) ? num2 : num1;

}

OUTPUT:

Enter any two numbers: 10 20

Maximum= 20

Minimum = 10

* #include<stdio.h>

// functions declaration

int add(int n1, int n2);

int subtract(int n1, int n2);

int multiply(int n1, int n2);

int divide(int n1, int n2);

// main function

int main()

{

int num1, num2;

printf("Enter two numbers: ");

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

printf("%d + %d = %d\n", num1, num2, add(num1, num2));

printf("%d - %d = %d\n", num1, num2, subtract(num1, num2));

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

printf("%d / %d = %d\n", num1, num2, divide(num1, num2));

return 0;

}

// function to add two integer numbers

int add(int n1, int n2)

{

int result;

result = n1 + n2;

return result;

}

// function to subtract two integer numbers

int subtract(int n1, int n2)

{

int result;

result = n1 - n2;

return result;

}

// function to multiply two integer numbers

int multiply(int n1, int n2)

{

int result;

result = n1 \* n2;

return result;

}

// function to divide two integer numbers

int divide(int n1, int n2)

{

int result;

result = n1 / n2;

return result;

}

Output:

Enter two numbers : 20 5

20+5 =25

20-5 =15

20\*5 =100

20/5=4

**f.**

* #include <stdio.h>

void main()

{

double number, sum = 0, i;

printf("**\n** enter the number ");

scanf("%lf", &number);

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

{

sum = sum + (1 / i);

if (i == 1)

printf("**\n** 1 +");

else if (i == number)

printf(" (1 / %lf)", i);

else

printf(" (1 / %lf) + ", i);

}

printf("**\n** The sum of the given series is %.2lf", sum);

}

Output:

$ cc pgm.c

$ a.out

enter the number 4

1 + (1/2.000000) + (1/3.000000) + (1/4.000000)

The sum of the given series is 2.08

/4.000000)

The sum of the given series is 2.08

* #include<stdio.h>

int findReverse(int n)

{

int sum=0;

while (n!=0)

{

sum = sum\*10 + n%10;

n /= 10;

}

return sum;

}

int main()

{

int number, reverse;

printf("Enter a positive interger: ");

scanf("%d", &number);

reverse = findReverse(number);

printf("The reverse of %d is: %d", number, reverse);

return 0;

}

Output:

Enter a positive integer: 12345

The reverse of 12345 is : 54321

**g.**

* # #include*<stdio.h> // include stdio.h library*

int fibonacci(int);

int main(void)

{

int terms;

printf("Enter terms: ");

scanf("%d", &terms);

**for**(int n = 0; n < terms; n++)

{

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

}

**return** 0; *// return 0 to operating system*

}

int fibonacci(int num)

{

*//base condition*

**if**(num == 0 || num == 1)

{

**return** num;

}

**else**

{

*// recursive call*

**return** fibonacci(num-1) + fibonacci(num-2);

}

}

Output:

Enter terms: 20

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181

* #include <stdio.h>

/\* Function declaration \*/

void printEvenOdd(int cur, int limit);

int main()

{

int lowerLimit, upperLimit;

// Input lower and upper limit from user

printf("Enter lower limit: ");

scanf("%d", &lowerLimit);

printf("Enter upper limit: ");

scanf("%d", &upperLimit);

printf("Even/odd Numbers from %d to %d are: ", lowerLimit, upperLimit);

printEvenOdd(lowerLimit, upperLimit);

return 0;

}

/\*\*

\* Recursive function to print even or odd numbers in a given range.

\*/

void printEvenOdd(int cur, int limit)

{

if(cur > limit)

return;

printf("%d, ", cur);

// Recursively call to printEvenOdd to get next value

printEvenOdd(cur + 2, limit);

}

Output:

Enter lower limit: 1

Enter upper limit: 100

Even/odd Numbers from 1 to 100 are: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99

**h.**

#include<stdio.h>

int main()

{

//fill the code;

int n;

scanf(“%d”,&n);

int arr[n];

int i;

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

{

scanf(“%d”,&arr[i]);

}

printf(“Reversed array is:\n”);

for(i = n-1; i >= 0; i–)

{

printf(“%d\n”,arr[i]);

}

return 0;

}

Output :

4 (size of the array)

2 (array elements)

6

7

1

Reversed array is :

1

7

6

2

* #include <stdio.h>

#include <math.h>

#define MAXSIZE 10

void main()

{

float x[MAXSIZE];

int i, n;

float average, variance, std\_deviation, sum = 0, sum1 = 0;

printf("Enter the value of N **\n**");

scanf("%d", &n);

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

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

{

scanf("%f", &x[i]);

}

*/\* Compute the sum of all elements \*/*

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

{

sum = sum + x[i];

}

average = sum / (float)n;

*/\* Compute variance and standard deviation \*/*

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

{

sum1 = sum1 + pow((x[i] - average), 2);

}

variance = sum1 / (float)n;

std\_deviation = sqrt(variance);

printf("Average of all elements = %.2f**\n**", average);

printf("variance of all elements = %.2f**\n**", variance);

printf("Standard deviation = %.2f**\n**", std\_deviation);

}

Output:

Enter the value of N

5

Enter 5 real numbers

34

88

32

12

10

Average of all elements = 35.20

variance of all elements = 794.56

Standard deviation = 28.19

**i.**

#include<stdio.h>

#include<conio.h>

int main() {

   int i, j, a[10][10], sum, rows, columns;

   printf("\nEnter the number of Rows : ");

   scanf("%d", &rows);

   printf("\nEnter the number of Columns : ");

   scanf("%d", &columns);

   //Accept the Elements in Matrix

   for (i = 0; i < rows; i++)

      for (j = 0; j < columns; j++) {

         printf("\nEnter the Element a[%d][%d] : ", i, j);

         scanf("%d", &a[i][j]);

      }

   //Addition of all Diagonal Elements

   sum = 0;

   for (i = 0; i < rows; i++)

      for (j = 0; j < columns; j++) {

         // Condition for Upper Triangle

         if (i < j) {

            sum = sum + a[i][j];

         }

      }

   //Print out the Result

   printf("\nSum of Upper Triangle Elements : %d", sum);

   return (0);

}

Output:

Enter the number of Rows : 3

Enter the number of Columns : 3

Enter the Element a[0][0] : 1

Enter the Element a[0][1] : 2

Enter the Element a[0][2] : 3

Enter the Element a[1][0] : 2

Enter the Element a[1][1] : 1

Enter the Element a[1][2] : 1

Enter the Element a[2][0] : 1

Enter the Element a[2][1] : 2

Enter the Element a[2][2] : 1

Sum of Upper Triangle Elements : 6

* #include<stdio.h>

void display(int result[], int n)

{

int i;

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

{

printf(“%d “, result[i]);

}

}

void maxi\_row(int mat[][3], int m, int n)

{

int i = 0, j;

int max = 0;

int result[m];

while (i < m)

{

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

{

if (mat[i][j] > max)

{

max = mat[i][j];

}

}

result[i] = max;

max = 0;

i++;

}

display(result, m);

}

int main()

{

int m, n;

scanf(“%d %d”,&m,&n);

int i, j;

int mat1[m][n];

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

{

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

scanf(“%d”,&mat1[i][j]);

}

maxi\_row(mat1,m,n);

return 0;

}

Output:

Enter the order of the matrix: 3 3

Input matrix elements :

1 2 3

4 5 6

7 8 9

3

6

9

**j.**

#include <string.h>

int main()

{

    char s[1000];

    int i,n,c=0;

    printf("Enter  the string : ");

    gets(s);

    n=strlen(s);

    for(i=0;i<n/2;i++)

    {

     if(s[i]==s[n-i-1])

     c++;

}

if(c==i)

    printf("string is palindrome");

    else

        printf("string is not palindrome");

    return 0;

}

Output :

Enter the string : Welcome

Before reverse : Welcome

After reverse : Emoclew

* #include <stdio.h>  
     
  void substring(char [], char[], int, int);  
     
  int main()  
  {  
     char string[1000], sub[1000];  
     int position, length, c = 0;  
     
     printf("Input a string**\n**");  
     gets(string);  
     
     printf("Enter the position and length of substring**\n**");  
     scanf("%d%d", &position, &length);  
     
     substring(string, sub, position, length);  
     
     printf("Required substring is **\"**%s**\"\n**", sub);  
     
     return 0;  
  }  
  *// C substring function definition*  
  void substring(char s[], char sub[], int p, int l) {  
     int c = 0;  
       
     while (c < l) {  
        sub[c] = s[p+c-1];  
        c++;  
     }  
     sub[c] = '**\0**';  
  }

Output:

Input a string

swayam

Enter the position and length of substring

a

Segmentation fault