**Q.1 Write a program in C to display the cube of the number up to a given integer.**

**Test Data :**

**Input number of terms : 5**

**Expected Output :**

**Number is : 1 and cube of the 1 is :1**

**Number is : 2 and cube of the 2 is :8**

**Number is : 3 and cube of the 3 is :27**

**Number is : 4 and cube of the 4 is :64**

**Number is : 5 and cube of the 5 is :125**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q1)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

int num;

printf("-->> Input Number of Terms : ");

scanf("%d",&num);

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

printf("--> Number is : %d and cube of the %d is : %d\n",i,i,(i\*i\*i));

}

return 0;

}

**Q.2 Write a program in C to display the multiplication table of a given integer.**

**Test Data :**

**Input the number (Table to be calculated) : 15**

**Expected Output :**

**15 X 1 = 15**

**...**

**...**

**15 X 10 = 150**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q2)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

int num;

printf("-->> Input the Number (Table to be Calculated) : ");

scanf("%d",&num);

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

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

}

return 0;

}

**Q.3 Write a program in C to display the multiplication table vertically from 1 to n.**

**Test Data :**

**Input upto the table number starting from 1 : 8**

**Expected Output :**

**Multiplication table from 1 to 8**

**1x1 = 1, 2x1 = 2, 3x1 = 3, 4x1 = 4, 5x1 = 5, 6x1 = 6, 7x1 = 7, 8x1 = 8**

**...**

**1x10 = 10, 2x10 = 20, 3x10 = 30, 4x10 = 40, 5x10 = 50, 6x10 = 60, 7x10 = 70, 8x10 = 80**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q3)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

int num;

printf("-->> Input Upto the Table Number Starting from 1 : ");

scanf("%d",&num);

printf("\nMultiplication Table from 1 to %d\n",num);

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

for(int j=1; j<=num; j++) {

printf("%dx%d = %d",j,i,(i\*j));

if(j<num) {

printf(", ");

}

}

printf("\n");

}

return 0;

}

**Q.4 Write a program in C to find the sum of the series [ 1-X^2/2!+X^4/4!- .........].**

**Test Data :**

**Input the Value of x :2**

**Input the number of terms : 5**

**Expected Output :**

**the sum = -0.415873**

**Number of terms = 5**

**value of x = 2.000000**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q4)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

#include<math.h>

double factorial(int n) {

if(n==0 || n==1) {

return 1;

}

else {

return n \* factorial(n-1);

}

}

int main() {

double x, sum=0;

int num;

printf("-->> Input the Value of x : ");

scanf("%lf",&x);

printf("-->> Input the Number of Terms : ");

scanf("%d",&num);

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

double term = pow(-1, i) \* pow(x, 2\*i) / factorial(2\*1);

sum += term;

}

printf("--> The Sum : %lf\n",sum);

printf("--> Number of Terms : %d\n",num);

printf("--> Value of x : %lf\n",x);

return 0;

}

**Q.5 Write a program in C to display the n terms of harmonic series and their sum.**

**1 + 1/2 + 1/3 + 1/4 + 1/5 ... 1/n terms**

**Test Data :**

**Input the number of terms : 5**

**Expected Output :**

**1/1 + 1/2 + 1/3 + 1/4 + 1/5 +**

**Sum of Series up to 5 terms : 2.283334**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q5)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

#include<math.h>

int main() {

int n;

double sum=0;

printf("-->> Input the Number of Terms : ");

scanf("%d",&n);

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

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

if(i<n) {

printf("+ ");

}

sum += 1.0/i;

}

printf("\n--> Sum of Series up to %d Terms : %.6lf\n",n,sum);

return 0;

}

**Q.6 Write a program in C to display the sum of the series [ 1+x+x^2/2!+x^3/3!+....].**

**Test Data :**

**Input the value of x :3**

**Input number of terms : 5**

**Expected Output :**

**The sum is : 16.375000**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q6)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

#include<math.h>

int factorial(int n) {

int fact = 1;

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

fact \*= i;

}

return fact;

}

int main() {

int terms;

double x,sum = 0.0;

printf("-->> Input the Value of x : ");

scanf("%lf",&x);

printf("-->> Input Number of Terms : ");

scanf("%d",&terms);

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

double term = pow(x,i) / factorial(i);

sum += term;

}

printf("--> The Sum is : %lf\n",sum);

return 0;

}

**Q.7 Write a program in C to find the sum of the series [ x - x^3 + x^5 + ......].**

**Test Data :**

**Input the value of x :2**

**Input number of terms : 5**

**Expected Output :**

**The values of the series:**

**2**

**-8**

**32**

**-128**

**512**

**The sum = 410**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q7)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

#include<math.h>

int main() {

double x;

int terms, sum=0;

printf("-->> Input the Value of x : ");

scanf("%lf",&x);

printf("-->> Input Number of Terms : ");

scanf("%d",&terms);

printf("--> The Value of the Series : \n");

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

int exp = 2 \* i + 1;

double term = pow(x,exp);

if(i%2 == 1) {

term = -term;

}

printf("%.0f\n",term);

sum += term;

}

printf("--> The Sum = %d\n",sum);

return 0;

}

**Q.8 Write a c program to check whether a given number is a perfect number or not.**

**Test Data:**

**Input the number: 56**

**Expected Output:**

**The positive divisor: 1 2 4 7 8 14 28**

**The sum of the divisor is: 64**

**So, the number is not perfect.**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q8)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

int n, sum=0;

printf("-->> Input the Number : ");

scanf("%d",&n);

printf("--> The Positive Divisor : ");

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

if(n%i == 0) {

printf("%d ",i);

sum += i;

}

}

printf("\n--> The Sum of the Divisor is : %d",sum);

if(sum == n) {

printf("\n--> So, the Number is Perfect.\n");

} else {

printf("\n--> So, the Number is not Perfect.\n");

}

return 0;

}

**Q.9 Write a c program to find the perfect numbers within a given number of ranges.**

**Test Data:**

**Input the starting range or number : 1**

**Input the ending range of number : 50**

**Expected Output :**

**The Perfect numbers within the given range : 6 28**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q9)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

int a, b;

printf("-->> Input the Starting Range or Number : ");

scanf("%d",&a);

printf("-->> Input the Ending Range or Number : ");

scanf("%d",&b);

printf("--> Perfect Number within the Given Range : ");

for(int n=a; n<=b; n++) {

int sum=0;

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

if(n%i == 0) {

sum += i;

}

}

if(sum == n) {

printf("%d ",n);

}

}

printf("\n");

return 0;

}

**Q.10 Write a C program to check whether a given number is an Armstrong number or not.**

**Test Data:**

**Input a number: 153**

**Expected Output:**

**153 is an Armstrong number.**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q10)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

#include<math.h>

int main() {

int n, originalNum, rem, result=0;

printf("-->> Input a Number : ");

scanf("%d",&n);

originalNum = n;

while(n != 0) {

rem = n % 10;

result += pow(rem,3);

n /= 10;

}

if(result == originalNum) {

printf("--> %d is an Armstrong Number\n",originalNum);

} else {

printf("--> %d is not an Armstrong Number\n",originalNum);

}

return 0;

}

**Q.11 Write a C program to find the Armstrong number for a given range of number.**

**Test Data :**

**Input starting number of range: 1**

**Input ending number of range : 1000**

**Expected Output :**

**Armstrong numbers in given range are: 1 153 370 371 407**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q11)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

#include<math.h>

int main() {

int a, b, originalNum, rem, n, result;

printf("-->> Input Starting Number of Range : ");

scanf("%d",&a);

printf("-->> Input Ending Number of Range : ");

scanf("%d",&b);

printf("--> Armstrong Numbers in Given Range are : ");

for(int i=a; i<=b; i++) {

originalNum = i;

result = 0;

n = 0;

while(originalNum != 0) {

rem = originalNum % 10;

result += pow(rem,3);

originalNum = originalNum / 10;

}

if(i == result) {

printf("%d ",i);

}

}

printf("\n");

return 0;

}

**Q.12 Write a program in C to display the first n terms of Fibonacci series.**

**Fibonacci series 0 1 2 3 5 8 13 .....**

**Test Data :**

**Input number of terms to display : 10**

**Expected Output :**

**Here is the Fibonacci series up to 10 terms :**

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

**CODE:**

*// Name: Angat Nayanbhai Shah (Q12)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

int t1 = 0, t2 = 1, t3;

int n;

printf("-->> Input Number of Terms to Display : ");

scanf("%d",&n);

printf("--> Here is the Fibonacci Series Up To %d terms : %d %d ",n,t1,t2);

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

t3 = t1 + t2;

printf("%d ",t3);

t1 = t2;

t2 = t3;

}

printf("\n");

return 0;

}

**Q.13 Write a program in C to display the number in reverse order. Test Data :**

**Input a number: 12345**

**Expected Output :**

**The number in reverse order is : 54321**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q13)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

int n, reverseNum=0, originalNum;

printf("-->> Input a Number : ");

scanf("%d",&n);

originalNum = n;

while(n != 0) {

int r = n % 10;

reverseNum = reverseNum \* 10 + r;

n /= 10;

}

printf("--> The Number in Reverse Order is : %d\n",reverseNum);

return 0;

}

**Q.14 Write a program in C to check whether a number is a palindrome or not.**

**Test Data:**

**Input a number: 121**

**Expected Output:**

**121 is a palindrome number.**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q14)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

int n, reverseNum=0, originalNum;

printf("-->> Input a Number : ");

scanf("%d",&n);

originalNum = n;

while(n != 0) {

int r = n % 10;

reverseNum = reverseNum \* 10 + r;

n /= 10;

}

if(originalNum == reverseNum) {

printf("--> %d is a Palindrome Number.\n", originalNum);

} else {

printf("--> %d is not a Palindrome Number.\n", originalNum);

}

return 0;

}

**Q.15 Write a program in C to find the number and sum of all integer between 100 and 200**

**which are divisible by 9.**

**Expected Output :**

**Numbers between 100 and 200, divisible by 9 :**

**108 117 126 135 144 153 162 171 180 189 198**

**The sum : 1683**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q15)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

int sum=0;

printf("--> Numbers Between 100 and 200, Divisible by 9 : \n");

for(int i=100; i<200; i++) {

if(i%9 == 0) {

printf("%d ",i);

sum += i;

}

}

printf("\n--> The Sum : %d\n",sum);

return 0;

}

**Q.16 Write a program in C to convert a decimal number into binary without using an array.**

**Test Data:**

**Enter a number to convert: 25**

**Expected Output :**

**The Binary of 25 is 11001.**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q16)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

void convertToBinary(int num) {

if(num > 1) {

convertToBinary(num/2);

}

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

}

int main() {

int decimalNum;

printf("-->> Enter a Number to Convert : ");

scanf("%d",&decimalNum);

printf("--> The Binary of (%d) is : ",decimalNum);

convertToBinary(decimalNum);

printf(".\n");

return 0;

}

**Q.17 Write a program in C to convert a binary number into a decimal number without using**

**array, function and while loop.**

**Test Data :**

**Input a binary number :1010101**

**Expected Output :**

**The Binary Number : 1010101**

**The equivalent Decimal Number : 85**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q17)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

long binaryNum;

int decimalNum=0, base=1, lastDigit;

printf("-->> Input a Binary Number : ");

scanf("%ld",&binaryNum);

printf("--> The Binary Number : %ld\n",binaryNum);

for(;binaryNum != 0; binaryNum/=10) {

lastDigit = binaryNum % 10;

decimalNum += lastDigit \* base;

base \*= 2;

}

printf("--> The equivalent Decimal Number : %d\n",decimalNum);

return 0;

}

**Q.18 Write a C program to find HCF (Highest Common Factor) of two numbers.**

**Test Data :**

**Input 1st number for HCF: 24**

**Input 2nd number for HCF: 28**

**Expected Output :**

**HCF of 24 and 28 is : 4**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q18)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

int num1, num2, temp;

printf("-->> Input 1st Number for HCF : ");

scanf("%d",&num1);

printf("-->> Input 2nd Number for HCF : ");

scanf("%d",&num2);

printf("--> HCF of %d and %d is : ",num1, num2);

while(num2 != 0) {

temp = num2;

num2 = num1 % num2;

num1 = temp;

}

printf("%d\n",num1);

return 0;

}

**Q.19 Write a program in C to find LCM of any two numbers.**

**Test Data :**

**Input 1st number for LCM: 15**

**Input 2nd number for LCM: 20**

**Expected Output :**

**The LCM of 15 and 20 is : 60**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q19)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int HCF(int a, int b) {

while(b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

int main() {

int num1, num2, lcm=0, hcf=0;

printf("-->> Input 1st Number for LCM : ");

scanf("%d",&num1);

printf("-->> Input 2nd Number for LCM : ");

scanf("%d",&num2);

hcf = HCF(num1, num2);

lcm = (num1 \* num2) / hcf;

printf("--> The LCM of %d and %d is : %d\n",num1,num2,lcm);

return 0;

}

**Q.20 Write a C program to find the length of a string without using the library function.**

**Test Data:**

**Input a string: welcome**

**Expected Output :**

**The string contains 7 number of characters.**

**So, the length of the string welcome is : 7**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q20)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include<stdio.h>

int main() {

char str[100];

int length = 0;

printf("-->> Input a String: ");

scanf("%s",&str);

while(str[length] != '\0') {

length++;

}

printf("--> The String contains %d Number of Characters.\n",length);

printf("--> So, the Length of the String %s is : %d\n",str,length);

return 0;

}

**Q.21 Write a C program to read a matrix A (MxN) and to find the following using**

**functions**

**a) Sum of the elements of each row**

**b) Sum of the elements of each column**

**c) Find the sum of all the elements of the matrix**

**Output the computed results with suitable headings.**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q21)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include <stdio.h>

void sumRows(int matrix[10][10], int rows, int cols) {

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

int rowSum = 0;

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

rowSum += matrix[i][j];

}

printf("--> Sum of Elements in Row %d: %d\n", i + 1, rowSum);

}

}

void sumColumns(int matrix[10][10], int rows, int cols) {

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

int colSum = 0;

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

colSum += matrix[i][j];

}

printf("--> Sum of Elements in Column %d: %d\n", j + 1, colSum);

}

}

int sumMatrix(int matrix[10][10], int rows, int cols) {

int totalSum = 0;

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

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

totalSum += matrix[i][j];

}

}

return totalSum;

}

int main() {

int matrix[10][10], rows, cols;

printf("-->> Enter the Number of Rows : ");

scanf("%d",&rows);

printf("-->> Enter the Number of Columns : ");

scanf("%d",&cols);

printf("-->>Enter the Elements of the Matrix :\n");

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

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

scanf("%d", &matrix[i][j]);

}

}

printf("\n");

sumRows(matrix, rows, cols);

printf("\n");

sumColumns(matrix, rows, cols);

printf("\n");

printf("--> Sum of all Elements in the Matrix: %d\n", sumMatrix(matrix, rows, cols));

return 0;

}

**Q.22 C Program to accept two matrices and check if they are equal or not?. Program will**

**accept the two matrices, and return true if their order and their elements are equal,**

**i.e. for all , if a[i][j]==b[i][j].**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q22)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include <stdio.h>

int matrixEqual(int a[10][10], int b[10][10], int rows, int cols) {

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

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

if(a[i][j] != b[i][j]) {

return 0;

}

}

}

return 1;

}

int main() {

int matrix1[10][10], matrix2[10][10], rows, cols;

printf("-->> Enter the Number of Rows : ");

scanf("%d", &rows);

printf("-->> Enter the Number of Columns : ");

scanf("%d", &cols);

printf("-->> Enter the Elements for Matrix 1.\n");

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

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

scanf("%d",&matrix1[i][j]);

}

}

printf("-->> Enter the Elements for Matrix 2.\n");

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

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

scanf("%d",&matrix2[i][j]);

}

}

if(matrixEqual(matrix1, matrix2, rows, cols) != 0) {

printf("--> Both the Matrices are Equal.\n");

} else {

printf("--> Both the Matrices are Not Equal.\n");

}

return 0;

}

**Q.23 C Program to check if a given matrix is an identity matrix or not. If I is theIdentity**

**Matrix,then for any matrix A, IA=AI=A. Program will check the given matrix is**

**identity or not, and prints the appropriate message.**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q23)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include <stdio.h>

int IdentityMatrix(int a[10][10], int size) {

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

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

if(i == j && a[i][j] != 1) {

return 0;

} else if(i != j && a[i][j] != 0) {

return 0;

}

}

}

return 1;

}

int main() {

int matrix[10][10], size;

printf("-->> Enter the Size of the Square Matrix: ");

scanf("%d",&size);

printf("-->> Enter the Elements of the %dx%d Matrix:\n",size,size);

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

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

scanf("%d",&matrix[i][j]);

}

}

if (IdentityMatrix(matrix, size)) {

printf("--> The Given Matrix is an Identity Matrix.\n");

} else {

printf("--> The Given Matrix is Not an Identity Matrix.\n");

}

return 0;

}

**Q.24 C program to find the frequency of odd numbers and even numbers in the input of a**

**matrix. Program will check the element type, if Matrix element is even, it ads 1 to**

**even counter otherwise add 1 to odd counter.**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q24)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include <stdio.h>

void countOddnEven(int a[10][10], int rows, int cols, int \*oddCount, int \*evenCount) {

\*oddCount = 0;

\*evenCount = 0;

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

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

if(a[i][j]%2 == 0) {

(\*evenCount)++;

} else {

(\*oddCount)++;

}

}

}

}

int main() {

int matrix[10][10], rows, cols, oddCount, evenCount;

printf("-->> Enter the Number of Rows : ");

scanf("%d",&rows);

printf("-->> Enter the Number of Columns : ");

scanf("%d",&cols);

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

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

scanf("%d",&matrix[i][j]);

}

}

countOddnEven(matrix, rows, cols, &oddCount, &evenCount);

printf("--> Frequency of Even Numbers: %d\n",evenCount);

printf("--> Frequency of Odd Numbers: %d\n",oddCount);

return 0;

}

**Q.25 C Program to interchange the main diagonal elements of the matrix. This Program**

**will accept a matrix of order M x N and store its elements and interchange the main**

**diagonal elements of the matrix with that of the secondary diagonal elements.**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q25)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include <stdio.h>

void interchangeDiagonals(int matrix[10][10], int size) {

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

int temp = matrix[i][i];

matrix[i][i] = matrix[i][size - i - 1];

matrix[i][size - i -1] = temp;

}

}

void display(int matrix[10][10], int size) {

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

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

printf("%d ", matrix[i][j]);

}

printf("\n");

}

}

int main() {

int matrix[10][10], size;

printf("-->> Enter the Size of the Square Matrix : ");

scanf("%d",&size);

printf("-->> Enter the Elements of the %dx%d Matrix:\n",size,size);

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

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

scanf("%d",&matrix[i][j]);

}

}

printf("--> Original Matrix.\n");

display(matrix, size);

interchangeDiagonals(matrix, size);

printf("--> Interchange Diagonals.\n");

display(matrix, size);

return 0;

}

**Q.26 C Program to sort the matrix rows and columns. This C program accept a order**

**MxN Matrix, and sort all rows of the matrix in ascending order and all columns in**

**descending order. In this program, we use the for statement to read two dimension**

**arrays.**

CODE:

*// Name: Angat Nayanbhai Shah (Q26)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include <stdio.h>

void sortRowAscending(int row[], int cols) {

for (int i=0; i<cols - 1; i++) {

for (int j=0; j<cols - i - 1; j++) {

if (row[j] > row[j + 1]) {

int temp = row[j];

row[j] = row[j + 1];

row[j + 1] = temp;

}

}

}

}

void sortColumnDescending(int matrix[10][10], int rows, int colIndex) {

for (int i=0; i<rows - 1; i++) {

for (int j=0; j<rows - i - 1; j++) {

if (matrix[j][colIndex] < matrix[j + 1][colIndex]) {

int temp = matrix[j][colIndex];

matrix[j][colIndex] = matrix[j + 1][colIndex];

matrix[j + 1][colIndex] = temp;

}

}

}

}

int main() {

int matrix[10][10], rows, cols;

printf("-->> Enter the Number of Rows : ");

scanf("%d",&rows);

printf("-->> Enter the Number of Columns : ");

scanf("%d",&cols);

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

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

scanf("%d",&matrix[i][j]);

}

}

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

sortRowAscending(matrix[i], cols);

}

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

sortColumnDescending(matrix, rows, j);

}

printf("--> Sorted Matrix According to the Condition.\n");

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

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

printf("%d ",matrix[i][j]);

}

printf("\n");

}

return 0;

}

**Q.27 C program to accept a matrix and determine whether it is a sparse matrix or not?. A**

**sparse matrix is a matrix, which has more zero elements than nonzero elements.**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q27)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include <stdio.h>

int sparseMatrix(int matrix[10][10], int rows, int cols) {

int zeroCount=0, total = rows \* cols;

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

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

if(matrix[i][j] == 0) {

zeroCount++;

}

}

}

if(zeroCount > total/2) {

return 1;

}

return 0;

}

int main() {

int matrix[10][10], rows, cols;

printf("-->> Enter the Number of Rows : ");

scanf("%d",&rows);

printf("-->> Enter the Number of Columns : ");

scanf("%d",&cols);

printf("-->> Enter the Elements for the Matrix : \n");

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

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

scanf("%d",&matrix[i][j]);

}

}

if(sparseMatrix(matrix, rows, cols)) {

printf("--> The Matrix is a Sparse Matrix.\n");

} else {

printf("--> The Matrix is Not a Sparse Matrix.\n");

}

return 0;

}

**Q.28 C Program to find the Inverse of a Matrix. To find the Matrix Inverse, matrix**

**should be a square matrix and Matrix Determinant is should not Equal to Zero. if A**

**is a Square matrix and |A|!=0, then AA’=I (I Means Identity Matrix).**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q28)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

#include <stdio.h>

int determinant(int matrix[10][10], int n) {

int det = 0;

if (n == 1) {

return matrix[0][0];

}

int temp[10][10];

int sign = 1;

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

int i=0, j=0;

for (int row=1; row<n; row++) {

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

if (col != f) {

temp[i][j++] = matrix[row][col];

if (j == n - 1) {

j = 0;

i++;

}

}

}

}

det += sign \* matrix[0][f] \* determinant(temp, n - 1);

sign = -sign;

}

return det;

}

void cofactor(int matrix[10][10], int temp[10][10], int row, int col, int n) {

int i = 0, j = 0;

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

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

if (r != row && c != col) {

temp[i][j++] = matrix[r][c];

if (j == n - 1) {

j = 0;

i++;

}

}

}

}

}

void adjacent(int matrix[10][10], int adj[10][10], int n) {

if (n == 1) {

adj[0][0] = 1;

return;

}

int sign = 1;

int temp[10][10];

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

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

cofactor(matrix, temp, i, j, n);

sign = ((i + j) % 2 == 0) ? 1 : -1;

adj[j][i] = sign \* determinant(temp, n - 1);

}

}

}

int inverse(int matrix[10][10], float inverse[10][10], int n) {

int det = determinant(matrix, n);

if (det == 0) {

printf("--> Inverse does not Exist. Determinant is Zero.\n");

return 0;

}

int adj[10][10];

adjacent(matrix, adj, n);

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

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

inverse[i][j] = adj[i][j] / (float)det;

}

}

return 1;

}

int main() {

int matrix[10][10], size;

float inv[10][10];

printf("-->> Enter the Size of the Square Matrix : ");

scanf("%d", &size);

printf("-->> Enter the Elements of the Matrix :\n");

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

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

scanf("%d", &matrix[i][j]);

}

}

if (inverse(matrix, inv, size)) {

printf("--> Inverse of the Matrix :\n");

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

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

printf("%.2f ", inv[i][j]);

}

printf("\n");

}

}

return 0;

}

**Q.29 Write c programs for following given patterns:**

**CODE:**

*// Name: Angat Nayanbhai Shah (Q29)*

*// Enrollment No: 202203103510097*

*// Branch: B.Tech CSE*

*#include<stdio.h>*

*int main() {*

*printf("<<------- PATTERN 1 ------->>\n");*

*for(int i=0; i<2; i++) {*

*for(int j=0; j<5; j++) {*

*printf("\* ");*

*}*

*printf("\n");*

*}*

*for(int i=0; i<3; i++) {*

*for(int j=0; j<5; j++) {*

*printf("\*");*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 2 ------->>\n");*

*for(int i=0; i<5; i++) {*

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

*printf("\* ");*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 3 ------->>\n");*

*for(int i=0; i<5; i++) {*

*for(int j=0; j<5-i; j++) {*

*printf(" ");*

*}*

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

*printf("\* ");*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 4 ------->>\n");*

*for(int i=5; i>0; i--) {*

*for(int j=0; j<5-i; j++) {*

*printf(" ");*

*}*

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

*printf("\* ");*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 5 ------->>\n");*

*for(int i=0; i<5; i++) {*

*for(int j=0; j<5-i; j++) {*

*printf("\* ");*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 6 ------->>\n");*

*for(int i=0; i<5; i++) {*

*for(int s=i; s<5; s++) {*

*printf(" ");*

*}*

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

*printf(" \*");*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 7 ------->>\n");*

*for(int i=0; i<5; i++) {*

*for(int j=0; j<4-i; j++) {*

*printf(" ");*

*}*

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

*printf("\* ");*

*}*

*printf("\n");*

*}*

*for(int i=4; i>0; i--) {*

*for(int j=0; j<5-i; j++) {*

*printf(" ");*

*}*

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

*printf("\* ");*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 8 ------->>\n");*

*int n8 = 1;*

*for(int i=0; i<5; i++) {*

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

*printf("%d ",n8);*

*n8++;*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 9 ------->>\n");*

*for(int i=0; i<6; i++) {*

*for(int j=0; j<6-i-1; j++) {*

*printf(" ");*

*}*

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

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

*}*

*printf("0 ");*

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

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

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 10 ------->>\n");*

*for(int i=0; i<5; i++) {*

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

*printf("%d ",i+1);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 11 ------->>\n");*

*for(int i=0; i<5; i++) {*

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

*printf("%d ",j+1);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 12 ------->>\n");*

*for(int i=0; i<5; i++) {*

*for(int j=0; j<5-i-1; j++) {*

*printf(" ");*

*}*

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

*printf("%d ",j+1);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 13 ------->>\n");*

*for(int i=0; i<7; i++) {*

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

*printf(" ");*

*}*

*for(char ch='A'; ch<='G'-i; ch++) {*

*printf("%c ",ch);*

*}*

*for(char ch='G'-i; ch>='A'; ch--) {*

*printf("%c ",ch);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 14 ------->>\n");*

*for(char i='A'; i<='C'; i++) {*

*for(char j='A'; j<='C'; j++) {*

*for(char k='A'; k<='C'; k++) {*

*printf("%c%c%c ",i,j,k);*

*}*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 15 ------->>\n");*

*for(int i=0; i<5; i++) {*

*for(int j=0; j<5-i; j++) {*

*printf("%d ",i+1);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 16 ------->>\n");*

*for(int i=4; i>0; i--) {*

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

*printf("%d ",j+1);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 17 ------->>\n");*

*for(int i=0; i<5; i++) {*

*for(int j=0; j<5; j++) {*

*if(j < i+1) {*

*printf("%d",5-i+j);*

*} else {*

*printf("5");*

*}*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 18 ------->>\n");*

*for(int i=0; i<5; i++) {*

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

*printf("%d",(j+1)%2);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 19 ------->>\n");*

*for(int i=0; i<4; i++) {*

*for(int j=1; j<=4-i; j++) {*

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

*}*

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

*printf("\*\*");*

*}*

*for(int j=4-i; j>0; j--) {*

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

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 20 ------->>\n");*

*for (int i=4; i>=0; i--) {*

*for (int j=5; j>0; j--) {*

*if (j == 5 - i) {*

*printf("\*");*

*} else {*

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

*}*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 21 ------->>\n");*

*for(int i=0; i<10; i++) {*

*for(int j=10-i; j<10; j++) {*

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

*}*

*printf("0");*

*for(int j=9; j>=10-i; j--) {*

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

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 22 ------->>\n");*

*for(int i=1; i<=5; i++) {*

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

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

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 23 ------->>\n");*

*for(int i=1; i<=5; i++) {*

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

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

*}*

*for (int j = 1; j <= 2 \* (5 - i) - 1; j++) {*

*printf(" ");*

*}*

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

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

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 24 ------->>\n");*

*for(int i=1; i<=4; i++) {*

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

*printf("%d",i);*

*if(i > j) {*

*printf("\*");*

*}*

*}*

*printf("\n");*

*}*

*for(int i=4; i>=1; i--) {*

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

*printf("%d",i);*

*if(i > j) {*

*printf("\*");*

*}*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 25 ------->>\n");*

*int num25 = 1;*

*for(int i=1; i<=4; i++) {*

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

*printf("%d",num25);*

*num25++;*

*}*

*int temp = num25 - 2;*

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

*printf("%d", temp);*

*temp--;*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 26 ------->>\n");*

*for(int i=0; i<4; i++) {*

*int start = 11 + i;*

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

*printf("%d ",start+j);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 27 ------->>\n");*

*int n27 = 1;*

*for(int i=0; i<5; i++) {*

*for(int s=i; s<5; s++) {*

*printf(" ");*

*}*

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

*printf("%3d ",n27);*

*n27++;*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 28 ------->>\n");*

*for(int i=1; i<=4; i++) {*

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

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

*}*

*for(int j=2; j<=i; j++) {*

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

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 29 ------->>\n");*

*int n29 = 1;*

*for(int i=5; i>0; i--) {*

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

*printf("%3d ",n29);*

*n29++;*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 30 ------->>\n");*

*for(int i=0; i<5; i++) {*

*int start = 1 + i;*

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

*printf("%d",start+j);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 31 ------->>\n");*

*for(int i=5; i>0; i--) {*

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

*printf("%d",i%2);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 32 ------->>\n");*

*for(int i=0; i<5; i++) {*

*for(int j=0; j<5; j++) {*

*printf("%d",(i+j)%4 + 1);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 33 ------->>\n");*

*for(int i=0; i<5; i++) {*

*for(int j=0; j<5; j++) {*

*if(i==0 || i==4 || j==0 || j==4) {*

*printf("1");*

*} else {*

*printf(" ");*

*}*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 34 ------->>\n");*

*int n34 = 1;*

*for(int i=0; i<5; i++) {*

*for(int s=i; s<5; s++) {*

*printf(" ");*

*}*

*for(int j=0; j<=i\*2; j++) {*

*printf("%3d ",n34\*n34);*

*n34++;*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 35 ------->>\n");*

*for(int i=0; i<5; i++) {*

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

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

*}*

*printf("\n");*

*}*

*for(int i=4; i>0; i--) {*

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

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

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 36 ------->>\n");*

*for(int i=0; i<5; i++) {*

*for(int s=i; s<5; s++) {*

*printf(" ");*

*}*

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

*printf("%d ",j+1);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 37 ------->>\n");*

*for(int i=0; i<4; i++) {*

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

*printf("0");*

*}*

*printf("\*");*

*for (int j=0; j<4-i-1; j++) {*

*printf("0");*

*}*

*printf("\*");*

*for (int j=0; j<4-i-1; j++) {*

*printf("0");*

*}*

*printf("\*");*

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

*printf("0");*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 38 ------->>\n");*

*int n38 = 4;*

*for(int i=1; i<=2\*n38-1; i++) {*

*for(int j=1; j<=2\*n38-1; j++) {*

*int min = i < j ? i : j;*

*if (min > 2 \* n38 - i) min = 2 \* n38 - i;*

*if (min > 2 \* n38 - j) min = 2 \* n38 - j;*

*printf("%d", n38 - min + 1);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 39 ------->>\n");*

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

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

*printf("%d ",i\*j);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 40 ------->>\n");*

*for (int i=0; i<5; i++) {*

*int value = 1;*

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

*printf("%d ",value);*

*value = value \* (i - j) / (j + 1);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 41 ------->>\n");*

*for(char i='E'; i>='A'; i--) {*

*for(char j=i; j<='E'; j++) {*

*printf("%c",j);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 42 ------->>\n");*

*for(char i='A'; i<='E'; i++) {*

*for(char j=i; j<='E'; j++) {*

*printf("%c",j);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 43 ------->>\n");*

*for(char i='A'; i<='E'; i++) {*

*for(char j='E'; j>=i; j--) {*

*printf("%c",j);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 44 ------->>\n");*

*for(char i='E'; i>='A'; i--) {*

*for(char j=i; j>='A'; j--) {*

*printf("%c",j);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 45 ------->>\n");*

*for(char i='E'; i>='A'; i--) {*

*for(char j='A'; j<=i; j++) {*

*printf("%c",i);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 46 ------->>\n");*

*for(char i='A'; i<='E'; i++) {*

*for(char j=i; j<='E'; j++) {*

*printf("%c",i);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 47 ------->>\n");*

*for(char i='A'; i<='E'; i++) {*

*for(char j='A'; j<=i; j++) {*

*printf("%c",j);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 48 ------->>\n");*

*for(char i='E'; i>='A'; i--) {*

*for(char j=i; j<='E'; j++) {*

*printf("%c",j);*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 49 ------->>\n");*

*int t1 = 1, t2 = 1, t3;*

*for(int i=0; i<5; i++) {*

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

*printf("%d ",t1);*

*t3 = t1 + t2;*

*t1 = t2;*

*t2 = t3;*

*}*

*printf("\n");*

*}*

*printf("\n<<------- PATTERN 50 ------->>\n");*

*for(int i=0; i<5; i++) {*

*for(int j=0; j<5; j++) {*

*if(i==0 || i==4 || j==0 || j==4) {*

*printf("1");*

*} else {*

*printf("0");*

*}*

*}*

*printf("\n");*

*}*

*return 0;*

*}*