// sum of rows and column of matrix

#include<stdio.h>

int main()

{

int A[3][3],i,j,row=0,column=0; // A is matrix of order 3 x 3

printf("Enter elements into the array : \n");

for(i=1;i<4;i++)

{

for(j=1;j<4;j++)

{

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

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

}

}

for(i=1;i<4;i++)

{

for(j=1;j<4;j++)

{

row=row+A[i][j];

}

printf("The sum of elements in row number %d is : %d\n",i,row);

row=0;

}

for(j=1;j<4;j++)

{

for(i=1;i<4;i++)

{

column=column+A[i][j];

}

printf("The sum of elements in column number %d is : %d\n",j,column);

column=0;

}

}

// sum of array

#include <stdio.h>

#define MAX\_SIZE 20

int sum(int arr[],int s,int len); // decleration

int main()

{

int arr[MAX\_SIZE];

int n,i,s;

printf("enter size of the array:");

scanf("%d",&n);

printf("enter element inside the array:");

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

{

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

}

s=sum(arr,0,n);

printf("sum of array elemnets:%d",s);

return 0;

}

int sum(int arr[],int s,int len)// recursion

{

if(s>=len)

return 0;

else

return (arr[s]+sum(arr,s+1,len));

}

// program to convert uppercase string to lowercase

#include<stdio.h>

#include<string.h>

int main(){

char str[30];

int i;

printf("Enter the string= ");//entering element in string form

scanf("%s",str);

for(i=0;i<=strlen(str);i++){ //converting uppercase to lowercase

if(str[i]>=65&&str[i]<=90)

str[i]=str[i]+32; //adding 32 will make the alphabet lowecase

}

printf("String in lower case= %s",str);

}

// tranpoose of a matrix

#include <stdio.h>

int main()

{

int matrix[20][20],transpose[20][20],r,c,i,j; /\*r is number of row,c is

number of column

trans is transpose\*/

printf("enter rows and column\n");

scanf("%d%d",&r,&c);

printf("enter elements of matrix\n");

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

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

{

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

}

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

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

transpose[j][i]=matrix[i][j];

printf("transpose of matrix: \n");

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

{

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

printf("%d\t",transpose[i][j]);

printf("\n");

}

return 0;

}

// inverse of a matrix

#include <stdio.h>

#include<string.h> //string.h is contains funcctions for manipulating arrays of character

#include<stdlib.h> //standard input/output library

#include<conio.h> //console input/output library

int i,j,n,k,size=0,r,c; // row,column

float num,determinent=0,inverse\_matrix[max][max],

matrix[max][max],new\_matrix[max][max],matrix\_minor[max][max],

matrix\_transpose[max][max];

float determinent(float matrix[max][max]);

float minor(float matrix[max][max],int k);

float transpose(float matrix[max][max]);

int main()

{

printf("\n degree of matrix :");

scanf("%d",&n);

printf("enetr elements of matrix: ");

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

{

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

{

printf("\n %d,%d : ",i,j);

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

}

printf("\n");

}

size=n;

determinent=determinent(matrix);

if(determinent=0)

{

printf(" \n inverse of this matrix does not exist");

getch();

return 0;

}

else

{

num=1/determinent;

matrix\_transpose[n][n]=transpose(matrix);

printf("\n\n matrix reverse is: \n\n");

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

{

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

{

inverse\_matrix[i][j]=num\*matrix\_transpose[i][j];

if(inverse\_matrix[i][j])

printf("%d",inverse\_matrix[i][j]);

else

printf("%d",inverse\_matrix[i][j]);

}

prinf("\n");

}

}

getch();

/\*We use a getch () function in a C/ C++ program to hold the output screen for some

time until the user passes a key from the keyboard to exit the console screen. Using getch ()

function, we can hide the input character provided by the users in the ATM PIN, password, etc.

Parameters: The getch () function does not accept any parameter from the user.\*/

return 0;

}

// caLCULATING determinent

float determinent(float matrix[][])

{

if(n==1)

return matrix[1][1];

else

{

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

{

determinent+=((-1)^(1+k))\*matrix[1][k]\*determinent

(minor9matrix,k));

n=size;

}

return determinent;

}

}

// caLCULATING minor

float minor(float matrix[][],int k)

{

int m=1,p,r,c,row=1,column;

column=k;

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

{

p=1;

for(c=1;c<=n;C++)

{

p=1;

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

{

if(r!=row &amp;amp; c!=column)

{

new\_matrix[m][p]=matrix[r][c];

p++;

}

if(r!=row)

m++;

}

}

}

n--;

return new\_matrix[m][p];

}

// caLCULATING transpose

float transpose(float matrix[][])

{

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

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

matrix\_transpose[i][j]=matrix[j][i];

return matrix\_transpose[n][n];

}

//palindrome string

/\*

Palindrome is a string, which when read in both

forward and backward way is same

To check if a string is a palindrome or not,

a string needs to be compared with the reverse of itself.\*/

#include <stdio.h>

#include <string.h>

int main()

{

char string1[20];

int i, length;

int flag = 0;

printf("Enter a string:");

scanf("%s", string1);

length = strlen(string1);

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

{

if(string1[i] != string1[length-i-1])

{

flag = 1;

break;

}

}

if (flag)

{

printf("%s is not a palindrome", string1);

}

else

{

printf("%s is a palindrome", string1);

}

return 0;

}

// maximum and minimuim in an array

#include <stdio.h>

int main()

{

int a[10],i,n,min,max;

printf("Enter size of the array(max=10) : "); // i have taken 5

scanf("%d",&n);

printf("Enter elements in array : ");

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

{

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

}

min=max=a[0];

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

{

if(min>a[i])

min=a[i];

if(max<a[i])

max=a[i];

}

printf("minimum of array is : %d",min);

printf("\nmaximum of array is : %d",max);

return 0;

}

//total number of alphabet and digits in a string

#include<stdio.h>

#define MAX\_SIZE 100

int main()

{

char str[MAX\_SIZE];

int alphabets,digits,i;

alphabets=digits=i=0;

printf("enter any string:");

gets(str);

while(str[i]!='\0')

{

if((str[i]>='a'&& str[i]<='z' ) || (str[i]>='A' && str[i]<='Z'))

{

alphabets++;

}

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

{

digits++;

}

i++;

}

printf("alphabets=%d\n",alphabets);

printf("digits=%d\n",digits);

return 0;

}

// number of words in a string

#include<stdio.h>

#include <string.h>

int main()

{

char s[1000];

int i,words=0;

printf("Enter the string : ");

gets(s);

for(i=0;s[i];i++)

{

if(s[i]==32)

words++;

}

if(i>0)

words++;

printf("no of words in string = %d\n",words);

return 0;

}

// product of two matrices

#include <stdio.h>

#define row 3

#define column 3

/\* Function declarations \*/

void matrix\_input(int mat[][column]);

void matrix\_print(int mat[][column]);

void matrix\_multiply(int mat1[][column], int mat2[][column], int res[][column]);

int main()

{

int mat1[row][column];

int mat2[row][column];

int product[row][column];

printf("enter elements for first matrix of size %d\*%d\n", row, column);

matrix\_input(mat1);

printf("enter elements for second matrix of size %d\*%d\n", row, column);

matrix\_input(mat2);

matrix\_multiply(mat1, mat2, product);

printf("product of both matrices is : \n");

matrix\_print(product);

return 0;

}

void matrix\_input(int mat[][column])

{

int r,c; // r is row, c is column

for (r = 0; r < row; r++)

{

for (c = 0; c < column; c++)

{

scanf("%d", (\*(mat + r) + c));

}

}

}

void matrix\_print(int mat[][column])

{

int r, c;

for (r = 0; r<row; r++)

{

for (c= 0; c < column; c++)

{

printf("%d ", \*(\*(mat + r) + c));

}

printf("\n");

}

}

void matrix\_multiply(int mat1[][column], int mat2[][column], int res[][column])

{

int r, c, i;

int sum;

for (r = 0; r < row; r++)

{

for (c = 0; c < column; c++)

{

sum = 0;

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

{

sum += (\*(\*(mat1 + r) + i)) \* (\*(\*(mat2 + i) + c));

}

\*(\*(res + r) + c) = sum;

}

}

}