Ex No:03 DAYS TO Months And days

Date:

Aim: Write a c program to convert days into months and days.

Algorithm:

Step 1: Start the execution.

Step2. Take the number of days as input.  
Step 3. For the number of years, divide the input by 365 and obtain its quotient.  
Step 4. For the number of weeks, divide the input by 365 and obtain its remainder.Further divide the remainder by 7(no of days in a week) and obtain its Step quotient.  
Step 5. For the number of days, divide the input by 365 and obtain its remainder.Further divide the remainder by 7(no of days in a week) and obtain its remainder.

Step 6: Stop the execution.

Program:-

#include <stdio.h>

#define DAYSINWEEK 7

void main()

{

int ndays, year, week, days;

printf("Enter the number of days\n");

scanf("%d", &ndays);

year = ndays / 365;

week =(ndays % 365) / DAYSINWEEK;

days =(ndays % 365) % DAYSINWEEK;

printf ("%d is equivalent to %d years, %d weeks and %d days",

ndays, year, week, days);

}

Output:-

Enter the number of days

1000

1000 is equivalent to 2 years, 38 weeks and 4 days

Result:-

Thus conversion of days into months and days program is executed successfully.

Ex No:07 PASCAL AND FLYDS TRIANGLE

Date:

Aim: Write a c program to create pascal and flyds triange.

Algorithm:

Step 1: Start the execution.

Step 2:Create pascal triangle using function and for loop.

Step 3:Create flyds triangle using function and for loop.

Step 4: Get input number of rows generated.

Step 5:Print pascal and flyds triangles.

Step 6:Stop execution.

Program:-

#include<stdio.h>

void pascaltri()

{

printf("PASCAL TRIANGLE\n");

int rows, coef = 1, space, i, j;

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

scanf("%d", &rows);

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

for (space = 1; space <= rows - i; space++)

printf(" ");

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

if (j == 0 || i == 0)

coef = 1;

else

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

printf("%4d", coef);

}

printf("\n");

}

}

void flydstri() {

printf("FLYDS TRIANGLE\n");

int rows, i, j, number = 1;

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

scanf("%d", &rows);

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

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

printf("%d ", number);

++number;

}

printf("\n");

}

}

int main()

{

pascaltri();

flydstri();

return 0;

}

Output:

PASCAL TRIANGLE

Enter the number of rows: 5

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

FLYDS TRIANGLE

Enter the number of rows: 5

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

Result:-

Thus pascal and flyds triangles are created successfully.

Ex No:08 TRACE OF MATRIX

Date:

Algorithm:-

Step 1:Start the execution

Step 2:Get input number of rows and columns of matrix

Step 3:Get input values of rows and columns of matrix

Step 4:Find and print trace of matrix value.

Step 5:Stop the execution.

Program:-

#include <stdio.h>

int main()

{

int i, j, rows, columns, trace = 0;

printf("Enter Matrix Rows and Columns = ");

scanf("%d %d", &rows, &columns);

int Tra\_arr[rows][columns];

printf("Please Enter the Matrix Items = \n");

i = 0;

while(i < rows)

{

j = 0;

while(j < columns)

{

scanf("%d", &Tra\_arr[i][j]);

j++;

}

i++;

}

i = 0;

while(i < rows)

{

j = 0;

while(j < columns)

{

if(i == j)

{

trace = trace + Tra\_arr[i][j];

}

j++;

}

i++;

}

printf("The Trace Of the Matrix = %d\n", trace);

}

Output:-

Enter Matrix Rows and Columns = 3

3

Please Enter the Matrix Items =

2

3

4

5

5

5

6

2

4

The Trace Of the Matrix = 11

Result:-

Thus find and trace of matrix program is executed successfully.

Ex No:09 ADDITION AND SUBTRACTION OF MATRIX

Date:

Aim:

Write a program for addition and subtraction of two matrix.

Algorithm:

Step 1: Start the execution.

Step 2: Get input number of rows and columns in two matrix.

Step 3: Get input values of two matrix.

Step 4:Find and print addition and subtraction of two matrix.

Step 5:Stop the execution.

Program:

#include <stdio.h>

#define MAXROW 10

#define MAXCOL 10

/\*User Define Function to Read Matrix\*/

void readMatrix(int m[][MAXCOL],int row,int col)

{

int i,j;

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

{

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

{

printf("Enter element [%d,%d] : ",i+1,j+1);

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

}

}

}

/\*User Define Function to Read Matrix\*/

void printMatrix(int m[][MAXCOL],int row,int col)

{

int i,j;

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

{

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

{

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

}

printf("\n");

}

}

int main()

{

int a[MAXROW][MAXCOL],b[MAXROW][MAXCOL],result[MAXROW][MAXCOL];

int i,j,r1,c1,r2,c2;

printf("Enter number of Rows of matrix a: ");

scanf("%d",&r1);

printf("Enter number of Cols of matrix a: ");

scanf("%d",&c1);

printf("\nEnter elements of matrix a: \n");

readMatrix(a,r1,c1);

printf("Enter number of Rows of matrix b: ");

scanf("%d",&r2);

printf("Enter number of Cols of matrix b: ");

scanf("%d",&c2);

printf("\nEnter elements of matrix b: \n");

readMatrix(b,r2,c2);

/\*sum and sub of Matrices\*/

if(r1==r2 && c1==c2)

{

/\*Adding two matrices a and b, and result storing in matrix result\*/

for(i=0;i< r1;i++)

{

for(j=0;j< c1;j++)

{

result[i][j]=a[i][j]+b[i][j];

}

}

/\*print matrix\*/

printf("\nMatrix after adding (result matrix):\n");

printMatrix(result,r1,c1);

/\*Subtracting two matrices a and b, and result storing in matrix result\*/

for(i=0;i< r1;i++)

{

for(j=0;j< c1;j++)

{

result[i][j]=a[i][j]-b[i][j];

}

}

/\*print matrix\*/

printf("\nMatrix after subtracting (result matrix):\n");

printMatrix(result,r1,c1);

}

else

{

printf("\nMatrix can not be added, Number of Rows & Cols are Different");

}

return 0;

}

Output:

Enter number of Rows of matrix a: 2

Enter number of Cols of matrix a: 2

Enter elements of matrix a:

Enter element [1,1] : 3

Enter element [1,2] : 4

Enter element [2,1] : 3

Enter element [2,2] : 5

Enter number of Rows of matrix b: 2

Enter number of Cols of matrix b: 2

Enter elements of matrix b:

Enter element [1,1] : 4

Enter element [1,2] : 6

Enter element [2,1] : 4

Enter element [2,2] : 6

Matrix after adding (result matrix):

7 10

7 11

Matrix after subtracting (result matrix):

-1 -2

-1 -1

Result:

Thus matrix of addition and subtraction program is executed successfully.

Ex No:12 FACTORIAL OF GIVEN NUMBERS

Date:

Aim:

Write a program to find factorial of given numbers.

Algorithm:

Step 1: Start the execution.

Step 2: Get input number to find factorial.

Step 3:Find and print factorial of given number.

Step 4:Stop the execution.

Program:

#include <stdio.h>

int main() {

int n, i;

unsigned long long fact = 1;

printf("Enter an integer: ");

scanf("%d", &n);

// shows error if the user enters a negative integer

if (n < 0)

printf("Error! Factorial of a negative number doesn't exist.");

else {

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

fact \*= i;

}

printf("Factorial of %d = %llu", n, fact);

}

return 0;

}

Output:

Enter an integer: 5

Factorial of 5 = 120

Result:

Thus factorial of given number program is executed successfully.

Ex No:13 ADDITION OF COMPLEX NUMBERS

Date:

Aim:

Write a program addition of complex numbers

Algorithm:

Step 1: Start the execution.

Step 2: Add the given complex numbers.

Step 3:print the addition of given complex number.

Step 4: Stop the execution.

Progrm:-

// C program to demonstrate

// addition of complex numbers

#include <stdio.h>

// define a structure for complex number

typedef struct complexNumber {

int real;

int img;

} complex;

// This function accepts two complex numbers

// as parameter and return addition of

// them.

complex add(complex x, complex y);

// Driver code

int main()

{

// Define three complex type numbers

complex a, b, sum;

// First complex number

a.real = 2;

a.img = 3;

// Second complex number

b.real = 4;

b.img = 5;

// Print first complex number

printf("\n a = %d + %di", a.real, a.img);

// print second complex number

printf("\n b = %d + %di", b.real, b.img);

// call add(a,b) function and

// pass complex numbers a & b

// as an parameter.

sum = add(a, b);

// Print result

printf("\n sum = %d + %di", sum.real, sum.img);

return 0;

}

// Complex add(complex x, complex y)

// function definition

complex add(complex x, complex y)

{

// Define a new complex number.

complex add;

// Add real part of a&b

add.real = x.real + y.real;

// Add Imaginary part of a&b

add.img = x.img + y.img;

return (add);

}

Output:

a = 2 + 3i

b = 4 + 5i

sum = 6 + 8i

Result:-

Thus addition of given complex number is executed successfully.

Ex No:15 PALINDROME OF STRING

Date:

Aim:

Write a program to check given string value is palindrome or not.

Algorithm:

Step 1: Start the execution.

Step 2: Get the input string value.

Step 3: Find reverse of string using string funtions.

Step 4: To check input string and reverse of given string is equal or not.

Step 5: If yes print the given string is palindrome ,otherwise print not palindrome

Step 4: Stop the execution.

Program:-

// C program to check palindromic string by reversing and

// comparing

#include <stdio.h>

#include <string.h>

int main()

{

char str[20];

printf("Enter String (upto 20 characters length): ");

scanf("%s", str);

int strSize = strlen(str);

if (strSize > 20) {

printf("\nError: strSize should be less than 20 "

"characters.");

return 0;

}

// reverse 'str' and store in 'strReversed'

char strReversed[20];

for (int i = strSize - 1; i >= 0; --i) {

strReversed[strSize - 1 - i] = str[i];

}

// comparing string, if they are equal then it's a

// palindromic string.

if (strcmp(str, strReversed) == 0) {

printf("\n%s is a palindrome", str);

}

else {

printf("\n%s is not a palindrome", str);

}

return 0;

}

Output

Enter String (upto 20 characters length): AMMA

AMMA is a palindrome

Enter String (upto 20 characters length): Flower

Flower is not a palindrome

Result:-

Thus to find the given string is palindrome or not program is executed successfully.