

### Algorithm :-

- Step 1: Start  
Step 2: declare required variables and initialize sum=0, rev=0  
Step 3: get the values for variable "num".  
Step 4: using while loop, perform step 5 until condition become false.  
Step 5: calculate  $d = \text{num} \% 10$ ,  
 $\text{num} = \text{num} / 10$   
 $\text{sum} = \text{sum} + d$   
 $\text{rev} = \text{rev} * 10 + d$   
Step 6: print the values of sum and reverse.  
Step 7: stop.

### Output :-

enter the number : 52  
sum of digits = 7  
Reverse of the number = 25

Prog. No. ....

Date. ....

Page No. ....

Aim :- Program to find the sum of digits and reverse of a number.

### SOURCE CODE :-

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    clrscr();
    int num, sum=0, rev=0, d;
    printf("Enter the number:");
    scanf("%d", &num);
    while (num) {
        d = num % 10;
        sum = sum + d;
        num = num / 10;
        rev = rev * 10 + d;
    }
    printf("Sum of digits = %d", sum);
    printf("Reverse of the number = %d", rev);
    getch();
}
```



### Algorithm :-

- Step 1 : Start  
Step 2 : Declare required variables  
Step 3 : Set the limit  $n$ ,  
Step 4 : Using for loop perform step 5 to until condition become false.  
Step 5 : Find fibonacci numbers by  
     $next = first + second$ ;  
     $first = second$ ;  
     $second = next$ ;  
Step 6 : Print n fibonacci numbers  
Step 7 : Stop.

Prog. No. ....

Date. ....

Page No. ....

Aim :- Find first  $n$  fibonacci numbers.

Source code :-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n, first = 0, second = 1, next, c;
    clrscr();
    printf("Enter the numbers of terms:");
    scanf("%d", &n);
    printf("First %d terms of Fibonacci series are :- \n", n);
    for (c = 0; c < n; c++)
    {
        if (c <= 1)
            next = c;
        else
            next = first + second;
        first = second;
        second = next;
    }
    printf("\n", next);
}
```

Output :-

Enter the number of terms : 5

First 5 terms of fibonacci series are:

0  
1  
1  
2  
3

```
}  
}  
getch();
```



### Algorithm :-

- Step 1: Start  
Step 2: declare required variables and initialize  $k=0$ .  
Step 3: get numbers of rows  
Step 4: using for loop "i" with initial value 0 and condition  $i \leq \text{rows}$  perform  
Step 5: using another for loop "space" with initial value 1 and space  $\leq \text{rows} - i$  perform Step 6.  
Step 6: perform print function, print (" ")  
Step 7: using while loop with  $k! = 2 * k - 1$ , print "\*" and placement k values.  
Step 8: print next line of pyramid by printf("\n")  
Step 9: stop.

### Output :-

Enter number of rows : 5

```

    *
  * *
* * *
* * *
* * *
```

Prog. No. ....  
Date .....

Page No. ....

Aim :- Program to create a pyramid using '\*'.

Source code :-

```
#include <stdio.h>
int main()
{
    int i, space, rows, k=0;
    printf("Enter number of rows:");
    scanf("%d", &rows);
    for (i=1; i<=rows; i++)
    {
        for (space=1; space<=rows-i; space++)
        {
            printf(" ");
        }
        while (k!=2*k-1)
        {
            printf("*");
            ++k;
        }
        printf("\n");
    }
    return 0;
}
```



### Algorithm :-

Step 1 : Start  
Step 2 : Read number  $n$   
Step 3 : Set  $f=0$   
Step 4 : for  $i=2$  to  $n-1$   
Step 5 : if  $n \bmod i = 0$  then  
Step 6 : set  $f=1$  and break.  
Step 7 : loop  
Step 8 : if  $f=0$  then  
print "The given number is prime"  
else  
print "The given number is not prime."  
Step 9 : Stop.

Prog. No. ....

Date. ....

Page No. ....

Aim :- Check whether a number is prime or not.

Source code :-

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    clrscr();
    int n, i, f=0;
    printf("Enter the number: ");
    scanf("%d", &n);
    for (i=2; i<n; i++)
    {
        if (n%i==0)
        {
            f=1;
        }
        break;
    }
    if (f==0)
        printf("The given number is prime");
    else
```

Output :-

Enter the number : 2  
The given number is prime.

Enter the number : 4  
The given number is not prime.

Prog. No.....

Date.....

Page No.....

```
printf("The given number is not prime");  
getch();  
}
```



### Algorithm :-

- Step 1 : Start  
Step 2 : declare functions like matrix input(), matrixprint(), matrix transpose() with attributes and variables.  
Step 3 : declare 2D matrix mat1[][ ]  
Step 4 : get values to mat1[][ ] using matrixinput().  
Step 5 : perform transpose of matrix using matrixtranspose with arguments mat1 and print the transpose of matrix mat1.  
Step 6 : Stop.

Prog. No.....  
Date.....

Page No.....

Aim :- Perform matrix transpose

Source code :-

```
#include <stdio.h>
#include <conio.h>
#define Row 2
#define Col 2
void matrixinput (int mat[][Col]);
void matrixprint (int mat[][Col]);
void matrixTranspose (int mat[][Col]);
void main ()
{
    int mat1[Row][Col];
    int mat2[Row][Col];
    int product[Row][Col], add[Row][Col],
        sub[Row][Col];
    clrscr();
    printf("Enter elements in matrix of\n size %d x %d\n", Row, Col);
    matrixinput(mat1);
    printf("\n matrix before transpose is\n");
    matrixprint(mat1);
    matrixTranspose(mat1);
    getch();
}
```



Output :-

Enter elements in matrix

of size 2x2

1  
2  
3  
4

matrix before transpose :

1 2  
3 4

matrix after transpose :

1 3  
2 4

Prog. No. ....  
Date: .....

Page No. ....

```
3
}
void matrixInput (int mat[][COL])
{
    int row, col;
    for (row=0; row<ROW; row++)
    {
        for (col=0; col<COL; col++)
        {
            scanf("%d", (*(mat+row)+col));
        }
    }
}

void matrix print (int, mat[][COL])
{
    int row, col;
    for (row=0; row<ROW; row++)
    {
        for (col=0; col<COL; col++)
        {
            printf("%d ", (*(mat+row)+col));
        }
        printf("\n");
    }
}

void matrix Transpose (int mat[][COL])
{
}
```



```
int row = 0, col, trans[Row][Col];
for (row = 0; row < Row; row++)
{
    for (col = 0; col < Col; col++)
    {
        *(&trans[col] + row) = *(&mat[row] + col);
    }
}

printf("matrix after transpose\n");
for (row = 0; row < Row; row++)
{
    for (col = 0; col < Col; col++)
    {
        printf("%d", *(&trans[row] + col));
    }
    printf("\n");
}
```



### Algorithm :-

- Step 1 : Start
- Step 2 : The user defined function denomination will divide the amount into 2000, 500, 100, 50, 20, 10, 5, 2, 1 rupees notes.
- Step 3 : Get the amount.
- Step 4 : Using for loop with limit size kind denomination of notes and denomination using formulas.
- Step 5 : Print denomination of given numbers.
- Step 6 : Stop

Prog. No. ....  
Date: .....

Page No. ....

Aim : Find the currency denomination of a given amount.

### SOURCE CODE :-

```
#include <stdio.h>
#include <conio.h>
#define size 9
void main ()
{
    int amount, notes, i;
    int denomination[size] = {2000, 500,
                                200, 100, 50, 20, 10, 5, 1};
    clrscr();
    printf("Enter amount:");
    scanf("%d", &amount);
    printf("%u\n", amount);
    for (i=0; i<size; i++)
    {
```

```
        notes = amount / denomination[i];
        if (notes)
```

```
    {
        amount = amount % denomination[i];
        printf("%d * %d = %d\n",
                notes, denomination[i],
```



Output :-

Enter amount : 254321

-3 \* 2000 = -6000  
-3 \* 500 = -1500  
-1 \* 200 = -200  
-1 \* 100 = -100  
-1 \* 20 = -20  
-3 \* 1 = -3

Prog. No.....  
Date.....

Page No.....

```
} notes * denomination[i]);  
getch();  
}
```

### Algorithm

- Step 1: Start and get a value for  
Step 2: declare "year"  
Variable  
Step 3: if  $\text{year} \% 4 == 0$ ,  $\text{year} \% 100 == 0$  is true  
and  $\text{year} \% 400 == 0$  is true  
then print leap year,  
else print not a leap year.  
Step 4: Stop.

Prog. No. \_\_\_\_\_

Date \_\_\_\_\_

Page No. \_\_\_\_\_

Aim: check for leap year.

Source code:-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int year;
    printf("Enter an year:");
    scanf("%d", &year);
    if (year % 4 == 0)
    {
        if (year % 100 == 0)
        {
            if (year % 400 == 0)
            {
                printf("%d is a leap year", year);
            }
            else
            {
                printf("%d is not a leap year", year);
            }
        }
        else
        {
            printf("%d is a leap year", year);
        }
    }
    else
    {
        printf("%d is not a leap year", year);
    }
}
```



Output :-

Enter an year : 2020  
2020 is an leap year.

```
}  
    printf("Year is not a leap year", year);  
    getch();  
}
```

### Algorithm :-

- Step 1 : start
- Step 2 : declare a char array S[200] and other required variables.
- Step 3 : get the string with initial value
- Step 4 : using for loop with initial value zero and condition  $S[i] != '\0'$ , perform Step 5.
- Step 5 : check condition  $S[i] == " "$ , if true perform  $Count++$ .
- Step 6 : print total words in given string
- Step 7 : stop.

### Output :-

Enter the string :  
welcome to yims  
Number of words in given string are : 3

Prog. No. ....  
Date .....

Page No. ....

Aim : Find the number of words in a sentence.

### Source code :-

```
#include <stdio.h>
#include <string.h>
void main()
{
    char s[200];
    int count = 0, i;
    printf("Enter the string : ");
    scanf("%s", s);
    for (i = 0; s[i] != '\0'; i++)
    {
        if (s[i] == ' ')
            count++;
    }
    printf("Number of words in given string are : %d", count + 1);
}
```