

This doc will be live till **31 July**. Students are advised to keep a copy of the program with them for future use

Index

1. WAP to check if a Number is Prime or Not. [Why use SqrRoot? : <https://tinyurl.com/yxms5r52>]
2. WAP to reverse a user entered number
3. Prime Add Number - <https://ibb.co/M9fCdfX>
4. Swap two numbers without using a third variable
5. Write a C program which will check whether a given number N is a Prime or Not. If the Number N is a Prime, then find its square root and print that value to the STDOUT as floating point number with exactly 2 decimal precision. If the number is not Prime, then print the value 0.00 to STDOUT.
6. Check if the user entered year is Leap year or not.
7. WAP to find n terms of the series 1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17, ...
8. Print the Armstrong numbers between the two intervals
9. MCQ on Pre and Post increment - <https://ibb.co/HHmYWj4>
10. Reverse string without built in function
11. Decimal to Octal
12. Write a program that will take a number in string format and convert it and print it in integer format. For e.g.,
13. Binary to Octal
14. GCD of three numbers
15. WAP to find all the Pythagorous Triplets between 2 limits
16. Check if a given number is a perfect square or not
17. Implement Matrix Multiplication Program
18. WAP to find the Second Largest Number in an Array
19. Add 2 user entered numbers and print the result in binary
20. Write a Program to remove vowels from a string?
21. Print the Pattern - <https://ibb.co/pbmZyNm>
22. Jumping Number - <https://ibb.co/WtNqTty>
23. WAP Program to Capitalise a String without using any inbuilt functions
24. Find the sum of series 1+4-9+16-25+36...
25. Find n terms of the series 3,3,9,9,15,27,21,81,27...
26. Palindrome Elements - <https://ibb.co/nsS3bZC>
27. Check if the user entered input is a **Harshad (Or Niven) Number**.
28. Write a program to multiply a number by 8 without using the multiply (*) operator.
29. Write a function that uses character handling library functions to determine the number of upper case, lower case, digits, spaces and punctuation characters in the specified text. (Reference - <https://tinyurl.com/y37fel3q>)
30. Print n terms of the following series - $n! + 1$

2	2	3	7	25	121	721...
$0!+1$	$1!+1$	$2!+1$	$3!+1$	$4!+1$	$5!+1$	$6!+1...$
31. Print the following pyramid
%
9%9
89%98
789%987
6789%9876
56789%98765

456789%987654

3456789%9876543

23456789%98765432

123456789%987654321

32. Write a function that removes all duplicate spaces from a sentence. Your program should ensure that only one space exists between words

33. Print matrix in snake pattern.

a. I/P

4

O/P

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

Explanation : i/p is 4 so a 4X4 matrix need to be used to do the program

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

And then traverse it in snake pattern

34. Print 'p' rows and 'q' columns of the following pattern

```
$*$*$*$*...
*$*$*$*$***...
*$*$*$*$***...
*$*$*$*$*...
*$*$*$*$***...
*$*$*$*$*...
*$*$*$*$***...
*$*$*$*$*...
*$*$*$*$*...
*$*$*$*$***...
.
.
.
```

35. Check whether a number is **Automorphic number** or not. If the number is not valid, it should display "Invalid input". (An automorphic number is a number whose square ends with the number itself.)For example, 5 is an automorphic number, $5*5 = 25$. The last digit is 5 which same as the given number.) $5^2 = 25$, $6^2 = 36$, $76^2 = 5776$, $376^2 = 141376$, and $890625^2 = 793212890625$, so 5, 6, 76, 376 and 890625 are all automorphic numbers.

36. Vote Counting Problem - <https://ibb.co/yXt81qJ>

37. Equilibrium Index - <https://ibb.co/LdVZsX7>

38. Check if the user entered input is a Strong Number or not.

39. Input 2 numbers - **num1** and **num2** - from the user

1. Detect if two integers have opposite signs. If same sign, return SAME else OPPOSITE
2. Add them without using any arithmetic operators and store the result in **result**
3. Convert the result into binary
4. Count the number of trailing zeros in the binary of **result** and store it in **count**
5. Invert all the binary bits of the **result** and store the decimal of the inverted binary as the **new_result**
6. Multiply **Count**+1 (#3+1) with the **new_result** (#4) and store it as **multiply**
7. Store **terms** number of subsequent cubes of **multiply** in a stack where **terms** is user entered

40. Super ASCII String question - <https://ibb.co/wYmqN6S>

- 41. Problem Statement : <https://ibb.co/fMr90sP>
 - a. Sample I/O : <https://ibb.co/Hz7MjFM>
- 42. String Rotation - <https://ibb.co/9HKYJth>
- 43.
- 44. X

1. WAP to check if a Number is Prime or Not. [Why use SqrRoot? : <https://tinyurl.com/yxms5r52>]

```
// Not the most efficient program
#include<stdio.h>
#include<math.h>
void isPrime(int num)
{
    int i,flag=0,root;
    for ( i=1 ; i<=num ; i++ )
        if (num%i==0)
            flag++;
    printf("%d is ", num);
    if(flag==2)
        printf("Prime\n");
    else
        printf("NOT Prime\n");
}

int main()
{
    int n;
    scanf("%d",&n);
    isPrime(n);
}
```

2. WAP to reverse a user entered number

```
#include<stdio.h>
int main()
{
    int num, new_num=0,rem;
    scanf("%d",&num); // say num is 397
    while(num!=0) // first iteration num is 397
    {
        rem=num%10; // in first iteration, will save 7 in rem
        new_num=new_num*10+rem; // new_num becomes 7
        num/=10; // num becomes 39
    } // loop continues with num as 39
    printf("%d\n", new_num); // Prints the final, reversed number
}
```

```
}
```

3. Prime Add Number - <https://ibb.co/M9fCdfX>

```
#include <stdio.h>

int isPrime(int num)
{
    int i, flag=0;

    for (i = 1; i<=num && i>0; i++)
    {
        if (num % i == 0)
        {
            ++flag;
        }
    }

    if (flag == 2)
        return 1;
    else
        return 0;
}

int main()
{
    //prime add
    int
num,temp_num,count=0,i=1,remainder,all_Prime_Add,odd_Prime_Add,even_Prime_Add;
    scanf("%d",&num);
    if (num>=0)
    {

        temp_num=num;
        while(temp_num!=0)
        {
            temp_num/=10;
            ++count;
        }

        temp_num=num;
```

```

while(i<=count)
{
    reminder=temp_num%10;
    all_Prime_Add+=reminder;
    if (i%2!=0)
    {
        odd_Prime_Add+=reminder;
    }
    else
    {
        even_Prime_Add+=reminder;
    }
    i++;
    temp_num/=10;
}

printf("Prime Add number :\t");
if (isPrime(all_Prime_Add)==1)
    printf("YES\n");
else
    printf("NO\n");

printf("Odd Prime Add number :\t");
if (isPrime(odd_Prime_Add)==1)
    printf("YES\n");
else
    printf("NO\n");

printf("Even Prime Add number :\t");
if (count<2)
    printf("INVALID\n");
else if (isPrime(even_Prime_Add)==1)
    printf("YES\n");
else
    printf("NO\n");
}
else
    printf("Invalid input\n");
}

```

4. Swap two numbers without using a third variable

```
#include <stdio.h>
```

```

int main()
{
    int x = 10, y = 5;
    // Code to swap 'x' and 'y'
    x = x + y; // x now becomes 15 y = x - y; // y becomes 10
    x = x - y; // x becomes 5
    printf("After Swapping: x = %d, y = %d", x, y); return 0;
}

```

5. Write a C program which will check whether a given number N is a Prime or Not. If the Number N is a Prime, then find its square root and print that value to the STDOUT as floating point number with exactly 2 decimal precision. If the number is not Prime, then print the value 0.00 to STDOUT.

First prog + sqrt function + printf("%.2f", num)

6. Check if the user entered year is Leap year or not.

```

#include <stdio.h>
int main()
{
    int year;
    printf("Enter a year: ");
    scanf("%d",&year);
    printf("\n");
    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);
    printf("\n");
    return 0;
}

```

7. WAP to find n terms of the series 1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17, ...

```
#include <stdio.h>
#define MAX 1000
void fibonacci(int n)
{
    int i, t1 = 0, t2 = 1, nextTerm;
    for (i = 1; i<=n; i++)
    { nextTerm = t1 + t2;
      t1 = t2;
      t2 = nextTerm; }
    printf("%d ", t1);
}
```

```
void prime(int n)
{
    int i, j, flag, count =0;
    for (i=2; i<=MAX; i++)
    { flag = 0;
      for (j=2; j<i; j++)
      { if(i%j == 0)
        { flag = 1;
          break;
        }
      }
      if (flag == 0)
      { if(++count == n)
        { printf("%d ", i);
          break;
        }
      }
    }
}
```

```
int main()
{
    int n,i=1;
    scanf("%d", &n);
    while(i<=n)
    { if(i%2 == 1)
      fibonacci (i/2 + 1);
      else
      prime(i/2);
      i++;
    }
    return 0;
}
```


8. Print the Armstrong numbers between the two intervals

```
#include <stdio.h>
#include <math.h>
int main()
{
    int start, end, i, temp1, temp2, remainder, n = 0, result = 0;

    printf("Enter start value and end value : ");
    scanf("%d %d", &start, &end);
    printf("\nArmstrong numbers between %d and %d are: ", start, end);

    for(i = start + 1; i < end; ++i)
    {
        temp2 = i;
        temp1 = i;

        while (temp1 != 0)
        {
            temp1 /= 10;
            ++n;
        }

        while (temp2 != 0)
        {
            remainder = temp2 % 10;
            result += pow(remainder, n);
            temp2 /= 10;
        }

        if (result == i)
            printf("%d ", i);

        n = 0;
        result = 0;
    }
    printf("\n");
    return 0;
}
```

9. MCQ on Pre and Post increment - <https://ibb.co/HHmYWj4>

a. Ans : C

10. Reverse string without built in function

```
#include<stdio.h>
#include<string.h>
```

```

int main() {
    char str[100], temp;
    int i, j = 0;

    printf("\nEnter the string :");
    scanf("%s", str);

    i = 0;
    j = strlen(str) - 1;

    while (i < j) {
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
        i++;
        j--;
    }

    printf("\nReverse string is :%s", str);
    return (0);
}

```

11. Decimal to Octal

Algorithm:

1. Store the remainder when the number is divided by 8 in an array.
2. Divide the number by 8 now
3. Repeat the above two steps until the number is not equal to 0.
4. Print the array in reverse order now.

```

#include <stdio.h>
#include <math.h>
int convertDecimalToOctal(int decimalNumber);
int main()
{
    int decimalNumber;
    printf("Enter a decimal number: ");
    scanf("%d", &decimalNumber);
    printf("%d in decimal = %d in octal", decimalNumber,
convertDecimalToOctal(decimalNumber));
    return 0;
}
int convertDecimalToOctal(int decimalNumber)
{
    int octalNumber = 0, i = 1;
    while (decimalNumber != 0)

```

```

{
    octalNumber += (decimalNumber % 8) * i;
    decimalNumber /= 8;
    i *= 10;
}
return octalNumber;
}

```

12. Write a program that will take a number in string format and convert it and print it in integer format. For e.g.,
 Input String => "574"
 Output Integer => 574

// C program to convert string to integer without using atoi function

```

#include <stdio.h>
int toString(char a[]);

int main()
{
    char a[100];
    int n;

    printf("Input a valid string to convert to integer\n");
    scanf("%s", a); // Whitespaces not allowed
    n = toString(a);

    printf("String = %s\nInteger = %d\n", a, n);

    return 0;
}

```

```

int toString(char a[]) {
    int c, sign, offset, n;

    if (a[0] == '-') // Handle negative integers
    {
        sign = -1;
    }

    if (sign == -1) // Set starting position to convert
    {
        offset = 1;
    }
    else
    {
        offset = 0;
    }
}

```

```

}

n = 0;

for (c = offset; a[c] != '\0'; c++)
{
    n = n * 10 + a[c] - '0'; // ASCII of 0 (zero) is 48
}

if (sign == -1) {
    n = -n;
}

```

13. Binary to Octal

```

#include <stdio.h>
#include <math.h>
int convertBinarytoOctal(long long binaryNumber);

int main()
{
    long long binaryNumber;
    printf("Enter a binary number: "); scanf("%lld", &binaryNumber);
    printf("%lld in binary = %d in octal", binaryNumber,
convertBinarytoOctal(binaryNumber) );
    return 0;
}

int convertBinarytoOctal(long long binaryNumber)
{
    int octalNumber = 0, decimalNumber = 0, i = 0;
    while(binaryNumber != 0)
    {
        decimalNumber += (binaryNumber%10) * pow(2,i); ++i;
        binaryNumber/=10;
    }
    i = 1;
    while (decimalNumber != 0)
    {
        octalNumber += (decimalNumber % 8) * i; decimalNumber /= 8;
        i *= 10;
    }
    return octalNumber;
}

```

14. GCD of three numbers

Definition of HCF (Highest common factor):

HFC is also called greatest common divisor (gcd). HCF of two numbers is a largest positive numbers which can divide both numbers without any remainder. For example HCF of two numbers 4 and 8 is 2 since 2 is the largest positive number which can dived 4 as well as 8 without a remainder.

Logic for writing program:

It is clear that any number is not divisible by greater than number itself. In case of more than one numbers, a possible maximum number which can divide all of the numbers must be minimum of all of that numbers.

For example: 10, 20, and 30

Min (10, 20, 30) =10 can divide all there numbers. So we will take one for loop which will start form min of the numbers and will stop the loop when it became one, since all numbers are divisible by one. Inside for loop we will write one if conditions which will check divisibility of both the numbers.

```
#include<stdio.h>
int gcd(int i , int j , int k);
int main()
{
    int i , j , k , g;
    scanf("%d %d %d", &i , &j , &k);
    g = gcd(i , j , k); printf("%d",g);
    return 0; }
int gcd(int i , int j , int k)
{
    int least;
    least = i;
    while(!( (i == j) && (j == k) ) )
    {
        i = (i == 0 ? least : i);
        j = (j == 0 ? least : j);
        k = (k == 0 ? least : k);
        if(i <= j)
        {
            if(i <= k)
                least = i;
            else
                least = k;
        }
        else
        {
            if(j <= k)
                least = j;
            else
                least = k;
        }
        i = i % least;
```

```

        j = j % least;
        k = k % least;
    }
    return least;
}

```

15. WAP to find all the Pythagorous Triplets between 2 limits

A Pythagorean triplet is a set of three integers a, b and c such that $a^2 + b^2 = c^2$.

```

#include<stdio.h>
int main(){
    int initial,final,a,b,c;
    printf("Enter the range in which you want to search for Pythagorean
Triplets:\nInitial (>0): ");
    scanf("%d",&initial);
    printf("\nFinal: ");
    scanf("%d",&final);
    printf("The Pythagorean Triplets in the given range are as follows:\n");
    for(a=initial;a<=final;a++){
        for(b=a;b<=final;b++){
            for(c=b;c<=final;c++){
                if(c*c==a*a+b*b){
                    printf("%d\t%d\t%d\n",a,b,c);
                }
            }
        }
    }
    return 0;
}

```

16. Check if a given number is a perfect square or not

```

#include <stdio.h>
#include <math.h>

int main()
{
    int num;
    int iVar;
    float fVar;

    printf("Enter an integer number: ");
    scanf("%d",&num);

    fVar=sqrt((double)num);
}

```

```

iVar=fVar;

if(iVar==fVar)
    printf("%d is a perfect square.",num);
else
    printf("%d is not a perfect square.",num);

return 0;
}

```

17. Implement Matrix Multiplication Program

```

#include <stdio.h>
int main()
{
    int a[10][10], b[10][10], result[10][10], r1, c1, r2, c2, i, j, k;
    printf("Enter rows and column for first matrix: ");
    scanf("%d %d", &r1, &c1);
    printf("Enter rows and column for second matrix: ");
    scanf("%d %d",&r2, &c2);
    // Column of first matrix should be equal to column of second matrix and
    while (c1 != r2)
    {
        printf("Error! column of first matrix not equal to row of
second.\n\n");
        printf("Enter rows and column for first matrix: ");
        scanf("%d %d", &r1, &c1);
        printf("Enter rows and column for second matrix: ");
        scanf("%d %d",&r2, &c2);
    }
    // Storing elements of first matrix.
    printf("\nEnter elements of matrix 1:\n");
    for(i=0; i<r1; ++i)
        for(j=0; j<c1; ++j)
        {
            printf("Enter elements a%d%d: ",i+1, j+1);
            scanf("%d", &a[i][j]);
        }
    // Storing elements of second matrix.
    printf("\nEnter elements of matrix 2:\n");
    for(i=0; i<r2; ++i)
        for(j=0; j<c2; ++j)
        {
            printf("Enter elements b%d%d: ",i+1, j+1);
            scanf("%d",&b[i][j]);
        }
    // Initializing all elements of result matrix to 0

```

```

for(i=0; i<r1; ++i)
    for(j=0; j<c2; ++j)
    {
        result[i][j] = 0;
    }
// Multiplying matrices a and b and
// storing result in result matrix
for(i=0; i<r1; ++i)
    for(j=0; j<c2; ++j)
        for(k=0; k<c1; ++k)
        {
            result[i][j]+=a[i][k]*b[k][j];
        }
// Displaying the result
printf("\nOutput Matrix:\n");
for(i=0; i<r1; ++i)
    for(j=0; j<c2; ++j)
    {
        printf("%d\t", result[i][j]);
        if(j == c2-1)
            printf("\n");
    }
return 0;
}

```

18. WAP to find the Second Largest Number in an Array
19. Add 2 user entered numbers and print the result in binary
20. Write a Program to remove vowels from a string?

```

#include <stdio.h>
#include <string.h>

int check_vowel(char);

int main()
{
    char s[100], t[100];
    int c, d = 0;

    printf("Enter a string to delete vowels\n");
    gets(s);

    for(c = 0; s[c] != '\0'; c++) {
        if(check_vowel(s[c]) == 0) {           // If not a vowel
            t[d] = s[c];
            d++;
        }
    }
}

```



```

}

t[d] = '\0';

strcpy(s, t); // We are changing initial string. This is optional.

printf("String after deleting vowels: %s\n", s);

return 0;
}

int check_vowel(char ch)
{
    if (ch == 'a' || ch == 'A' || ch == 'e' || ch == 'E' || ch == 'i' || ch == 'I' || ch == 'o' || ch == 'O' || ch == 'u' || ch == 'U')
        return 1;
    else
        return 0;
}

```

21. Print the Pattern - <https://ibb.co/pbmZyNm>

```

#include <stdio.h>
void printPattern(int);
void main()
{
    int N;
    scanf("%d",&N);
    if(N>=4 && N<=10)
        printPattern(N);
}

void printPattern(int N)
{
    int i,j;
    // Loop to print upper half of the pattern
    for(i=1; i<=N; i++)
    {
        for(j=i; j<=N; j++)
            printf("*");
        for(j=1; j<i; j++)
            printf(" ");
        for(j=i; j<=N; j++)
            printf("*");
        printf("\n");
    }
}

```

```
// Loop to print lower half of the pattern
```

```
for(i=1; i<=N; i++)
{
    for(j=1; j<=i; j++)
        printf("*");
    for(j=i-1; j<N-1; j++)
        printf(" ");
    for(j=1; j<=i; j++)
        printf("*");
    printf("\n");
}
}
```

22. Jumping Number - <https://ibb.co/WtNqTty>

```
#include <stdio.h>
int check_jump(int num);
int main()
{
    int num,i;

    scanf("%d",&num);

    for (i = 0; i <= num; ++i)
    {
        if (check_jump(i)==1)
            printf("%d ",i);
    }

    return 0;
}
int check_jump(int num)
{
    int flag=0,temp,temp_num,no_of_digits=0,nd;
    temp_num=num;
    while(temp_num!=0)
    {
        temp_num/=10;
        no_of_digits+=1;
    }

    temp_num=num;
    nd=no_of_digits;
    while(nd>1)
    {
```

```

temp=temp_num%10;
temp_num/=10;

if (temp - temp_num%10 == 1 || temp - temp_num%10 == -1)
{
    flag++;
}
else
{
    break;
}
--nd;
}

if(flag==no_of_digits-1)
    return 1;
else
    return 0;
}

```

23. WAP Program to Capitalise a String without using any inbuilt functions

```

#include <stdio.h>
#include <string.h>

int main()
{
    char Str1[100];
    int i;

    printf("\n Please Enter a String that you want to Convert into Uppercase : ");
    gets(Str1);

    for (i = 0; Str1[i]!='\0'; i++)
    {
        if(Str1[i] >= 'a' && Str1[i] <= 'z')
        {
            Str1[i] = Str1[i] -32;
        }
    }

    printf("\n The given String in Upper Case = %s", Str1);

    return 0;
}

```

```
}
```

24. Find the sum of series $1+4-9+16-25+36\dots$

```
#include<stdio.h>
#include<math.h>

int main()
{
    int a=1,i,n;
    printf("enter the number\n");
    scanf("%d",&n);
    for(i=1;i<n;i++)
    {
        a=a+pow(-1,i+1)*pow(i+1,2);
    }
    printf("sum is %d",a);
    return 0;
}
```

25. Find n terms of the series 3,3,9,9,15,27,21,81,27... (Hint : GP and AP)

```
#include <stdio.h>
int main()
{
    int num,i,ap_num=3,gp_num=3;
    scanf ("%d",&num);

    for (i = 1; i <= num; ++i)
    {
        if (i%2!=0)
        {
            printf("%d\t", ap_num);
            ap_num+=6;
        }
        else
        {
            printf("%d\t", gp_num);
            gp_num*=3;
        }
    }
}
```

```
}  
}
```

26. Palindrome Elements - <https://ibb.co/nsS3bZC>

```
// Function to check if palindrome or not  
int isPalindrome(int N) {  
    int res=0,rem=0,sum=0;  
    int temp =N;  
    while(N!=0){  
        rem = N%10;  
        sum = sum*10+rem;  
        N= N/10;  
    }  
    if(sum==temp){  
        res = 1;  
    }  
    return res;  
}  
  
// Function to check  
// if an array is PalinArray or not  
int isPalinArray(int arr[], int n) {  
    // Traversing each element of the array  
    // and check if it is palindrome or not  
    for (int i = 0; i < n; i++) {  
        int ans = isPalindrome(arr[i]);  
        if (ans == 0) {  
            return 0;  
        }  
    }  
    return 1;  
}  
  
int main() {  
    int N,*arr1;  
    scanf("%d",&N);  
    if(N>=2 && N<=100) {  
        int arr[N];  
        for(int i=0;i<N;i++){  
            scanf("%d",&arr[i]);  
        }  
        int res = isPalinArray(arr, N);  
        if (res == 1)  
            printf("Array is a PalinArray");  
        else  
            printf("Array is not a PalinArray");  
    }
```

```

    }
    return 0;
}

```

27. Check if the user entered input is a **Harshad (Or Niven) Number**. (A number divisible by the sum of its digits)

a. First few Harshad numbers : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 18, 20

```

// C program to check whether a number is a Harshad number or not
#include<stdio.h>
int main()
{
//fill the code
int num;
int temp;
scanf("%d",&num);
int sum = 0;
temp = num;
while(temp)
{
sum += temp % 10;
temp = temp / 10;
}
int res = num % sum;
if(res == 0)
printf("Harshad Number");
else
printf("Not Harshad Number");
return 0;
}

```

28. Write a program to multiply a number by 8 without using the multiply (*) operator.

```

#include <stdio.h>
int main()
{
    int num;
    scanf("%d",&num);
    printf("%dx8=%d\n",num,num<<3); //Left Shift 3 times to multiply by 8
}

```

29. Write a function that uses character handling library functions to determine the number of upper case, lower case, digits, spaces and punctuation characters in the specified text. (Reference - <https://tinyurl.com/y37fel3q>)

30. Print n terms of the following series - $n! + 1$

2	2	3	7	25	121	721...
$0!+1$	$1!+1$	$2!+1$	$3!+1$	$4!+1$	$5!+1$	$6!+1...$

31. Print the following pyramid

```
%
9%9
89%98
789%987
6789%9876
56789%98765
456789%987654
3456789%9876543
23456789%98765432
123456789%987654321
```

32. Write a function that removes all duplicate spaces from a sentence. Your program should ensure that only one space exists between words

33. Print matrix in snake pattern.

a. I/P

4

O/P

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

Explanation : i/p is 4 so a 4X4 matrix need to be used to do the program

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

And then traverse it in snake pattern

34. Print 'p' rows and 'q' columns of the following pattern

```
$*$*$*$*...
*$*$*$*$***...
*$*$*$*$***...
*$*$*$*$*...
*$*$*$*$***...
*$*$*$*$*...
*$*$*$*$***...
```

\$\$*\$*\$...
\$*\$*\$*\$*\$...
\$\$*\$*\$*\$...
\$\$*\$*\$***...

.
. .
.

35. Check whether a number is **Automorphic number** or not. If the number is not valid, it should display "Invalid input". (An automorphic number is a number whose square ends with the number itself.)For example, 5 is an automorphic number, $5*5 = 25$. The last digit is 5 which same as the given number.)
 $5^2 = 25$, $6^2 = 36$, $76^2 = 5776$, $376^2 = 141376$, and $890625^2 = 793212890625$, so 5, 6, 76, 376 and 890625 are all automorphic numbers.
36. Vote Counting Problem - <https://ibb.co/yXt81gJ>
37. Equilibrium Index - <https://ibb.co/LdVZsX7>
38. Check if the user entered input is a **Strong Number** or not. (Strong numbers are the numbers whose sum of factorial of digits is equal to the original number.)
 $145 = 1! + 4! + 5!$
 $40585 = 4! + 0! + 5! + 8! + 5!$
39. Input 2 numbers - **num1** and **num2** - from the user
8. Detect if two integers have opposite signs. If same sign, return SAME else OPPOSITE
 9. Add them without using any arithmetic operators and store the result in **result**
 10. Convert the result into binary
 11. Count the number of trailing zeros in the binary of **result** and store it in **count**
 12. Invert all the binary bits of the **result** and store the decimal of the inverted binary as the **new_result**
 13. Multiply **Count**+1 (#3+1) with the **new_result** (#4) and store it as **multiply**
 14. Store **terms** number of cubes of **multiply** in a stack where **terms** is user entered
40. X

Some helpful links

1. Undefined behaviour of pre and post increment
 - a. <https://stackoverflow.com/questions/949433/why-are-these-constructs-using-pre-and-post-increment-undefined-behavior>
2. Why use -lm during terminal compilation in Linux when header file *math.h* is used?
 - a. <https://askubuntu.com/questions/332884/how-to-compile-a-c-program-that-uses-math-h>
 - b. <https://stackoverflow.com/questions/10059146/gcc-libm-not-working/10059154#10059154>
- 3.