



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# C PROGRAMMING INTRODUCTION

## TUẦN 7: VÒNG LẶP

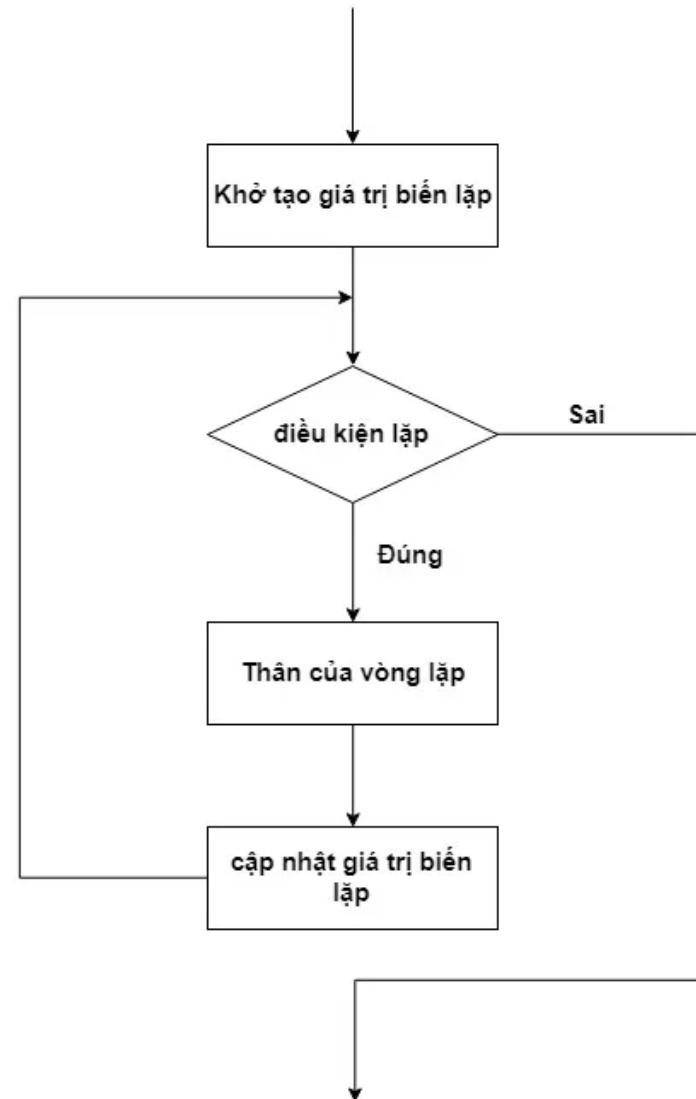
# Nội dung

- Vòng lặp For
  - Ôn tập
    - Cấu trúc lặp
    - Các chú ý
  - Thực hành

# Vòng lặp for

**for** ( *khởi tạo* ;  
điều kiện tiếp tục ;  
*cập nhật* )

câu lệnh



# Cấu trúc lặp **for**

- Định dạng cấu trúc lặp **for**  
**for** ( *khởi tạo* ; điều kiện tiếp tục ; *cập nhật* )  
câu lệnh
- Ví dụ: In các số nguyên từ 1 đến 10

```
for( int counter = 1; counter <= 10; counter++ )  
    printf( "%d\n", counter );
```

Không có  
dấu ; ở cuối

# Cấu trúc lặp `for` (II)

- Khởi tạo và cập nhật

- Các câu lệnh khởi tạo, cập nhật: cách nhau bởi dấu ,

```
for (int i = 0, j = 0; j + i <= 10; j++, i++)  
    printf( "%d\n", j + i );
```

**`for`** ( *khởi tạo* ; điều kiện tiếp tục ; *cập nhật* )  
 câu lệnh

# Cấu trúc For: Các chú ý

- Biểu thức số học:
  - khởi tạo, điều kiện tiếp tục và cập nhật có thể chứa các biểu thức số học.
  - Ví dụ: nếu  $x = 2$  và  $y = 10$   
`for ( j = x; j <= 4 * x * y; j += y / x )`  
tương đương với  
`for ( j = 2; j <= 80; j += 5 )`
- "cập nhật" có thể theo chiều âm (giảm)
- Nếu điều kiện tiếp tục có giá trị khởi tạo **false**
  - Thân của cấu trúc **for** sẽ không được thực hiện
  - Các câu lệnh sau cấu trúc **for** được thực hiện

# Cấu trúc For: Các chú ý (II)

- Biến điều khiển
  - Thường được in hoặc sử dụng bên trong thân vòng **for** nhưng không bắt buộc

```
for(int counter = 1; counter <= 10; counter++)  
    printf( "%d\n", counter );
```

**for** ( *khởi tạo* ; điều kiện tiếp tục ; *cập nhật* )  
 câu lệnh

# Ví dụ 1.

```
for (i=1; i<=100;i++) {  
    x += i;  
    if ((x % i) == 0) { i--; }  
}  
  
print("x = %d", x);  
print("i = %d", i)
```



# Ví dụ 2

```
char s[10] = "ABCDE";  
  
for (i=0, j=strlen(s)-1; i<j; i++,j--)  
{  
    c = s[i];  
    s[i] = s[j];  
    s[j] = c;  
}  
  
printf("%s", s);
```

# Ví dụ 3

```
char c;  
int count;  
for (count=0; (c=getchar()) != '\n'); count++)  
    { }  
printf("Number of characters is %d\n", count);
```

# Bài tập 7.1

- Viết chương trình in ra mười số nguyên và bình phương của chúng

1 1

2 4

3 9

...

10 100

# Lời giải

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

```
int main()
```

```
{
```

```
    int i;
```

```
    for(i = 1; i <= 10; i = i + 1)
```

```
        printf("%d %d\n", i, i * i);
```

```
    return 0;
```

```
}
```

# Bài tập 7.2

- Viết chương trình in ra tam giác như sau:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

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

```
int main()
```

```
{
```

```
    int i, j;
```

```
    for(i = 1; i <= 10; i = i + 1)
```

```
    {
```

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

```
            printf("*");
```

```
            printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

# Lời giải

# Bài tập 7.3

- Viết chương trình in ra số lượng các số lớn hơn 27 trong các số từ 1 đến 100

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

# Lời giải

```
int main()
```

```
{
```

```
    int i;
```

```
    int count = 0;
```

```
    for(i = 1; i <= 100; i = i + 1)
```

```
    {
```

```
        if (i > 27)
```

```
            count = count + 1;
```

```
    }
```

```
    printf("%d numbers were greater than 27\n", count);
```

```
    return 0;
```

```
}
```



# Bài tập 7.4

- Viết chương trình in ra các số nguyên tố  $\leq 100$
- Sử dụng thư viện `math.h` để dùng các hàm toán học: `sqrt`,...

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

```
main()
{
    int i, j;

    printf("%d\n", 2);

    for (i = 3; i <= 100; i = i + 1)
    {
        for (j = 2; j < i; j = j + 1)
        {
            if (i % j == 0)
                break;
```

# Lời giải

```
        if (j > sqrt(i))
        {
            printf("%d\n", i);
            break;
        } // end if (j
    } // end for (i=2 ...)
} // end for (i=3 ...)
return 0;
}
```

# Bài tập 7.5

- Sửa bài tập 7.4 bằng cách bỏ đi các số chẵn để không phải gọi hàm sqrt nhiều lần.

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

```
int main()
{
    int i, j;
    double sqrti;

    printf("%d\n", 2);

    for (i = 3; i <= 100; i = i + 2)
    {
        sqrti = sqrt(i);
```

# Lời giải

```
        for (j = 2; j < i; j = j + 1)
        {
            if(i % j == 0)
                break; /* not prime */
            if(j > sqrt(i))
            {
                /* prime */
                printf("%d\n", i);
                break;
            }
        } // end of for (j=2 ...
    } // end of for (i=3 ...)
    return 0;
}
```

```
/* Counting down to blast-off */
```

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

```
int main(void)
```

```
{
```

```
    int time, start;
```

```
    printf("Enter starting time (an integer) in seconds> ");
```

```
    scanf("%d", &start);
```

```
    printf("\nBegin countdown\n");
```

```
    for (time = start; time > 0; time = time - 1)
```

```
    {
```

```
        printf("T - %d\n", time);
```

```
    }
```

```
    printf("Blast-off!\n");
```

```
    return (0);
```

```
}
```

# Bài tập 7.6

# Bài tập 7.7

- Viết chương trình in ra bảng chuyển đổi nhiệt độ Celsius sang Fahrenheit.
- Chú ý điều kiện tiếp tục vòng lặp và sử dụng `#define` để định nghĩa hằng số
- $\text{fahrenheit} = 1.8 * \text{celsius} + 32.0;$

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

```
/* Constant macros */
```

```
#define CBEGIN 10
```

```
#define CLIMIT -5
```

```
#define CSTEP 5
```

```
int main(void)
```

```
{
```

```
    /* Variable declarations */
```

```
    int    celsius;
```

```
    double fahrenheit;
```

```
    /* Print the table heading */
```

```
    printf(" Celsius Fahrenheit\n");
```

# Lời giải

# Lời giải

```
/* Print the table */  
for (celsius = CBEGIN;  
    celsius >= CLIMIT;  
    celsius = celsius + CSTEP)  
{  
    fahrenheit = 1.8 * celsius + 32.0;  
    printf(" %3d   %7.2f\n", celsius, fahrenheit);  
}  
  
return (0);  
}
```



# Bài tập 7.8

- Gõ chương trình và biên dịch để hiểu cơ chế vòng lặp lồng nhau (vòng lặp ở bên trong vòng lặp)

# Bài 7.8

```
#include <stdio.h>
int main(void)
{
    int i, j;                /* loop control variables */

    printf("      I   J\n"); /* prints column labels */

    for (i = 1; i < 4; i = i + 1) /* heading of outer for loop */
    {
        printf("Outer %6d\n", i);
        for (j = 0; j < i; j = j + 1) /* heading of inner loop */
        {
            printf("  Inner%9d\n", j);
        } /* end of inner loop */
    } /* end of outer loop */
    return (0);
}
```

# Bài tập 7.9

- Viết chương trình sử dụng cấu trúc for để tính  $n$  giai thừa
- Test cases:

## Results

Enter n: 4

4! = 24

## Results

Enter n: 0

0! = 1

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

```
int main ()
```

```
{
```

```
    int i, n, f;
```

```
    printf (" Enter n: ");
```

```
    scanf ("%d", &n);
```

```
    f = 1;                                /* 0! */
```

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

```
        f *= i;                          /* Now , f = i! */
```

```
    }
```

```
    printf ("%d! = %d\n", n, f);
```

```
    return 0;
```

```
}
```

# Lời giải

# Bài tập 7.10

- Trong số học, số hoàn hảo là số nguyên dương có giá trị bằng tổng các ước số nguyên của nó (không kể chính nó). E.g:  $6=1+2+3$
- Viết chương trình in ra các số hoàn hảo có giá trị  $\leq N$  với  $N$  được người dùng nhập vào từ bàn phím

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

# Lời giải

```
void main()
```

```
{
```

```
    int n, i, j, tong;
```

```
    printf("\nEnter N= : ");
```

```
    scanf("%d", &n);
```

```
    for (i=2; i<=n; i++)
```

```
    {
```

```
        tong = 1;
```

```
        for (j=2; j<=i/2; j++)
```

```
            if (i%j == 0)
```

```
                tong += j;
```

```
            if (tong == i)
```

```
                printf("\n%d", i);
```

```
        }
```

```
    }
```

# Bài tập 7.11

The first 10 natural numbers are



1, 2, 3, 4, 5, 6, 7, 8, 9, 10

# Bài tập 7.12

The first 10 natural numbers are



1, 2, 3, 4, 5, 6, 7, 8, 9, 10



Sum of the first 10 natural numbers are

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55$$



# Bài tập 7.13

The first **n** natural number

$$n = 7$$

1, 2, 3, 4, 5, 6, 7

$$1 + 2 + 3 + 4 + 5 + 6 + 7 = 28$$

Sum of Natural Number : **28**

# Bài tập 7.14

1, 2, 3, 4, 5, 6, 7, 8, 9, 10



The sum of 10 no. is

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 \\ = 55$$



The Average is

$$55 / 10 = 5.5$$

**Note :-** Numbers may be any numbers.

# Bài tập 7.15

- Write a program in C to display the cube of the number upto a given integer.



Cube of the number is

Number	Cube
1	$1^3 = 1$
2	$2^3 = 8$
3	$3^3 = 27$
4	$4^3 = 64$
5	$5^3 = 125$

# Bài tập 7.16

- Write a program in C to display the multiplication table of a given integer.



15

Multiplication Table of 15



# Bài tập 7.17

- Write a program in C to display the pattern like right angle triangle with a number.

- 1
- 12
- 123
- 1234



```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10
```

# Bài tập 7.18

- Write a program in C to display the n terms of odd natural number and their sum like:

1 3 5 7 ... n

Number of terms is : 10

The odd numbers are



1, 3, 5, 7, 9, 11, 13, 15, 17, 19



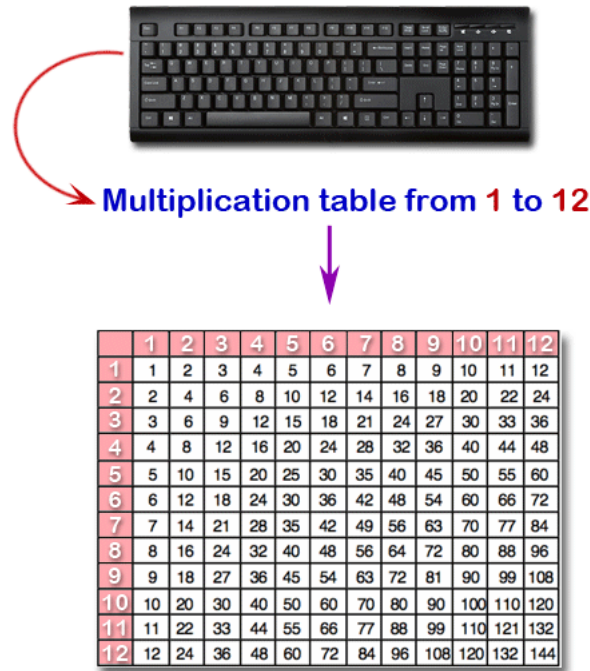
1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 + 17 + 19  
= 100



The Sum of odd Natural Number  
upto 10 terms is : 100

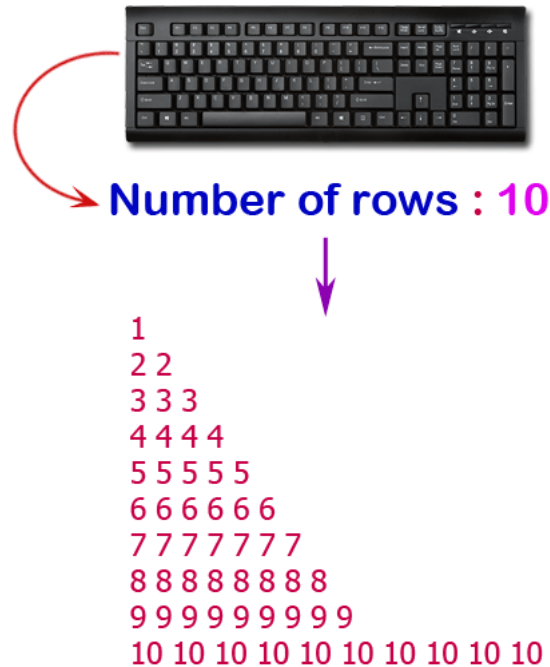
# Bài tập 7.19

- Write a program in C to display the multiplication table vertically from 1 to n.



# Bài tập 7.20

- Write a program in C to make such a pattern like right angle triangle with a number which will repeat a number in a row.





# Bài tập 7.21

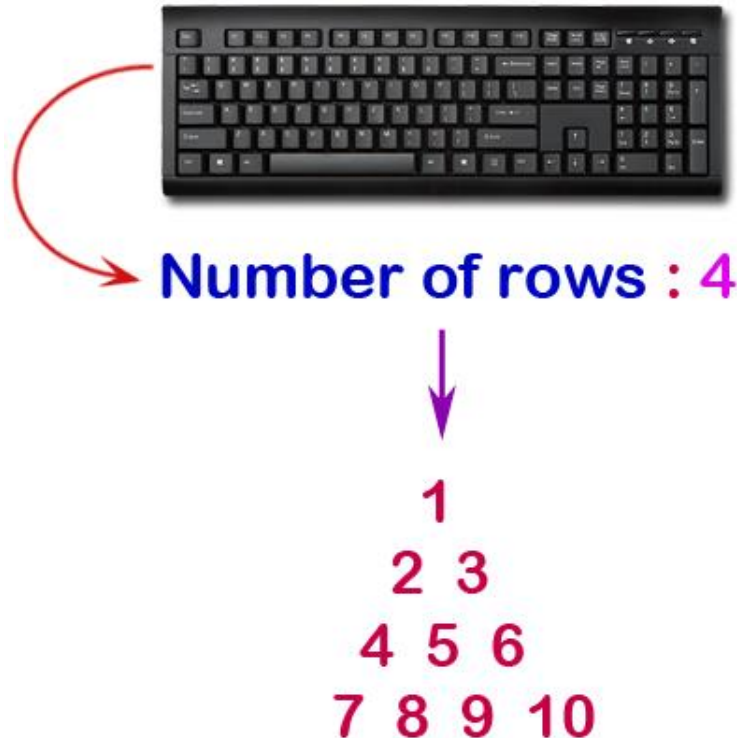
- Write a program in C to make such a pattern like right angle triangle with number increased by 1.



1  
2 3  
4 5 6  
7 8 9 10

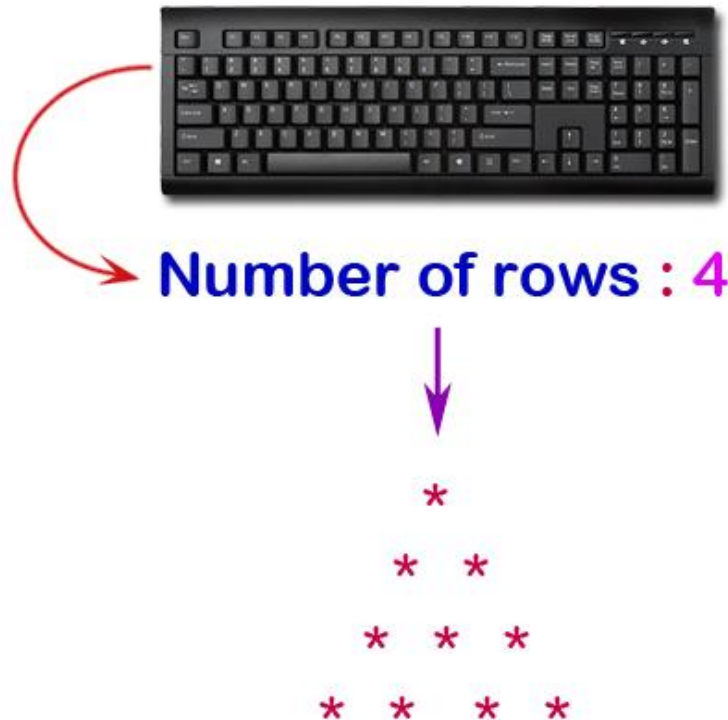
# Bài tập 7.22

- Write a program in C to make such a pattern like a pyramid with numbers increased by 1.



# Bài tập 7.23

- Write a program in C to make such a pattern like a pyramid with an asterisk.



# Bài tập 7.24

- Write a C program to calculate the factorial of a given number.

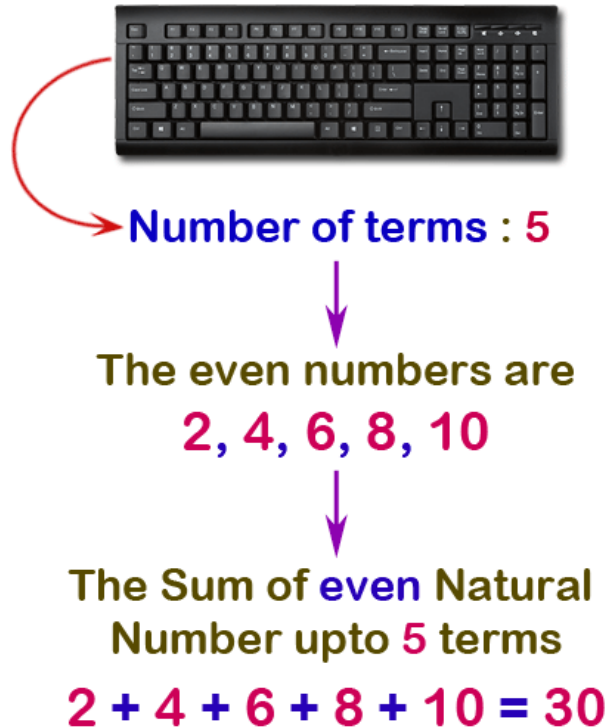


The Factorial of 5 is

$$5 \times 4 \times 3 \times 2 \times 1 = 120$$

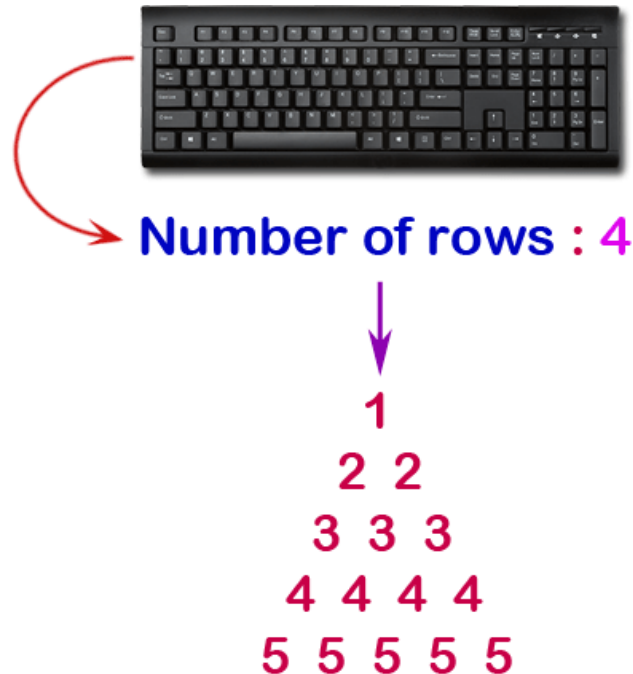
# Bài tập 7.25

- Write a program in C to display the n terms of even natural number and their sum.



# Bài tập 7.26

- Write a program in C to make such a pattern like a pyramid with a number which will repeat the number in the same row.



# Bài tập 7.27

- Write a program in C to find the sum of the series [  $1 - X^2/2! + X^4/4! - \dots$  ].
- X và số phần tử nhập từ bàn phím.

# Bài tập 7.28

- Write a program in C to display the n terms of harmonic series and their sum. The series is :  
 $1 + 1/2 + 1/3 + 1/4 + 1/5 \dots 1/n$  terms



**Number of terms : 5**



$$1/1 + 1/2 + 1/3 + 1/4 + 1/5$$

**Sum of Series upto 5 terms : 2.283334**



# Bài tập 7.29

- Write a program in C to display the sum of the series [  $9 + 99 + 999 + 9999 \dots$  ].



**Number of terms : 5**

**9, 99, 999, 9999, 99999**

**Sum of the series**

$$9 + 99 + 999 + 9999 + 99999 = 111105$$

# Bài tập 7.30

- Write a program in C to print the Floyd's Triangle.



1  
0 1  
1 0 1  
0 1 0 1  
1 0 1 0 1

# Bài tập 7.31

- Write a program in C to find the sum of the series [ $x - x^3 + x^5 + \dots$ ].
- X và số mục trong tổng: nhập từ bàn phím.

# Bài tập 7.32

- Write a program in C to display the n terms of square natural number and their sum. The series is as below:  
1 4 9 16 ... n Terms

# Bài tập 7.33

- Write a program in C to find the sum of the series  $1 + 11 + 111 + 1111 + \dots$  n terms.



Number of terms : 5

1, 11, 111, 1111, 11111

The sum is

$1 + 11 + 111 + 1111 + 11111 = 12345$

# Bài tập 7.34

- Write a c program to find the perfect numbers within a given number of range.
- Perfect number is a positive number which sum of all positive divisors excluding that number is equal to that number.
- For example 6 is perfect number since divisor of 6 are 1, 2 and 3.
- Sum of its divisor is  $1 + 2 + 3 = 6$

# Bài tập 7.35

- Write a C program to check whether a given number is an armstrong number or not.
- When the sum of the cube of the individual digits of a number is equal to that number, the number is called Armstrong number. For Example 153 is an Armstrong number because  $153 = 1^3 + 5^3 + 3^3$ .
- Test Data :  
Input a number: 153  
Expected Output :  
153 is an Armstrong number.

*Armstrong Number :*

Number = 153

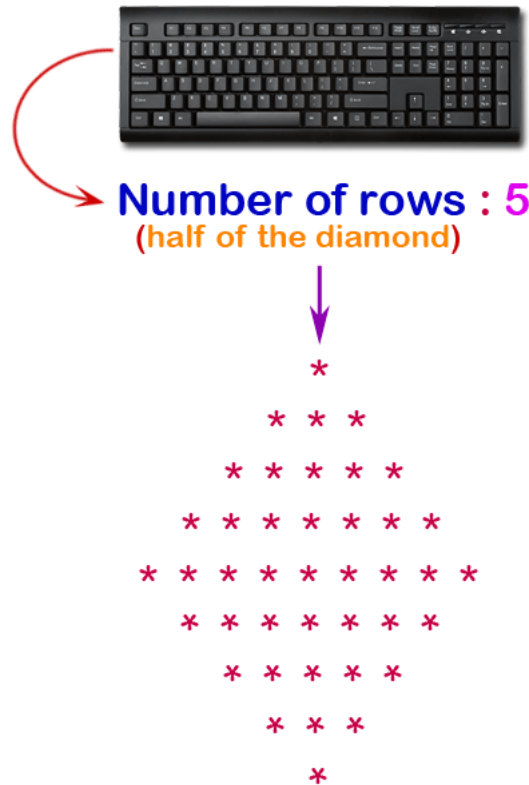
$$1^3 + 5^3 + 3^3$$
$$1 + 125 + 27 = 153$$

Sum = Original Number

153 is Armstrong Number

# Bài tập 7.36

- Write a program in C to display the pattern like a diamond.





# Bài tập 7.37

- Write a C program to display Pascal's triangle

Number of rows : 5



1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1

Pascal's triangle

# Bài tập 7.38

- Write a program in C to find the prime numbers within a range of numbers.
- Test case:

Input starting number of range: 1

Input ending number of range : 50

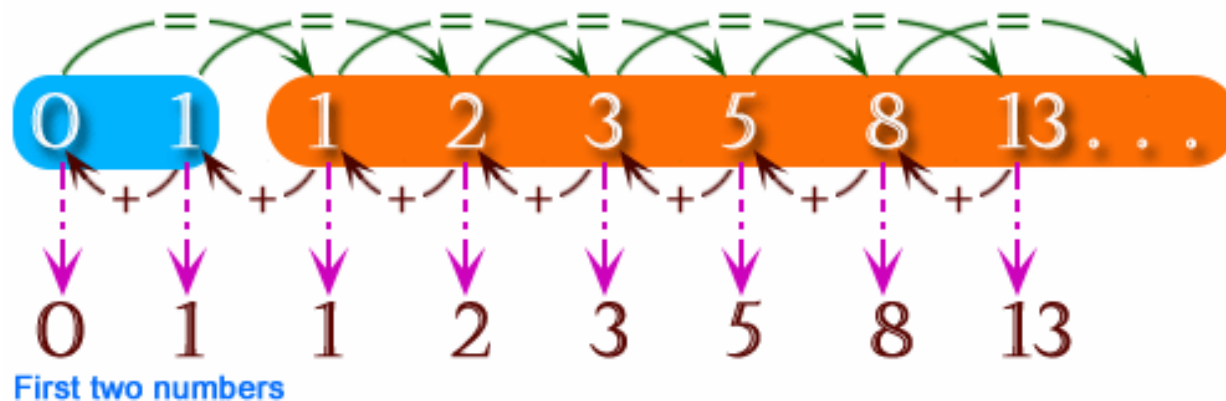
The prime numbers between 1 and 50 are :

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

# Bài tập 7.39

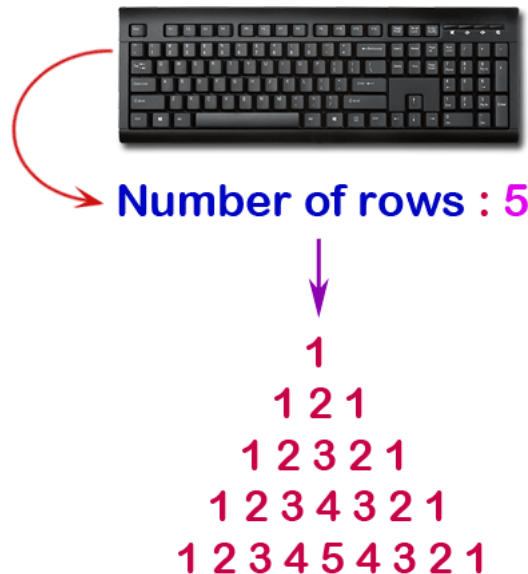
- Write a program in C to display the first n terms of Fibonacci series. The series is as follows:
- Fibonacci series 0 1 2 3 5 8 13 .....

*Fibonacci Sequence :*



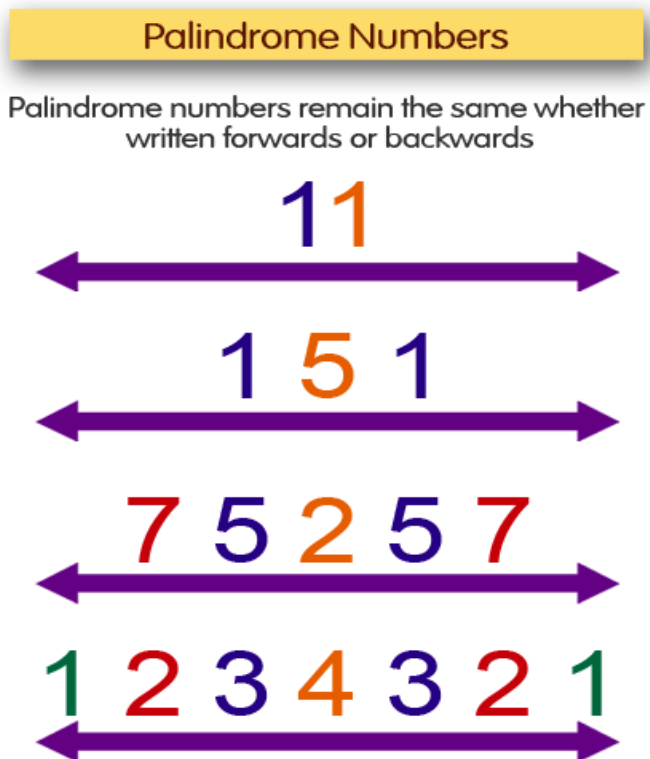
# Bài tập 7.40

- Write a program in C to display the such a pattern for n number of rows using a number which will start with the number 1 and the first and a last number of each row will be 1.



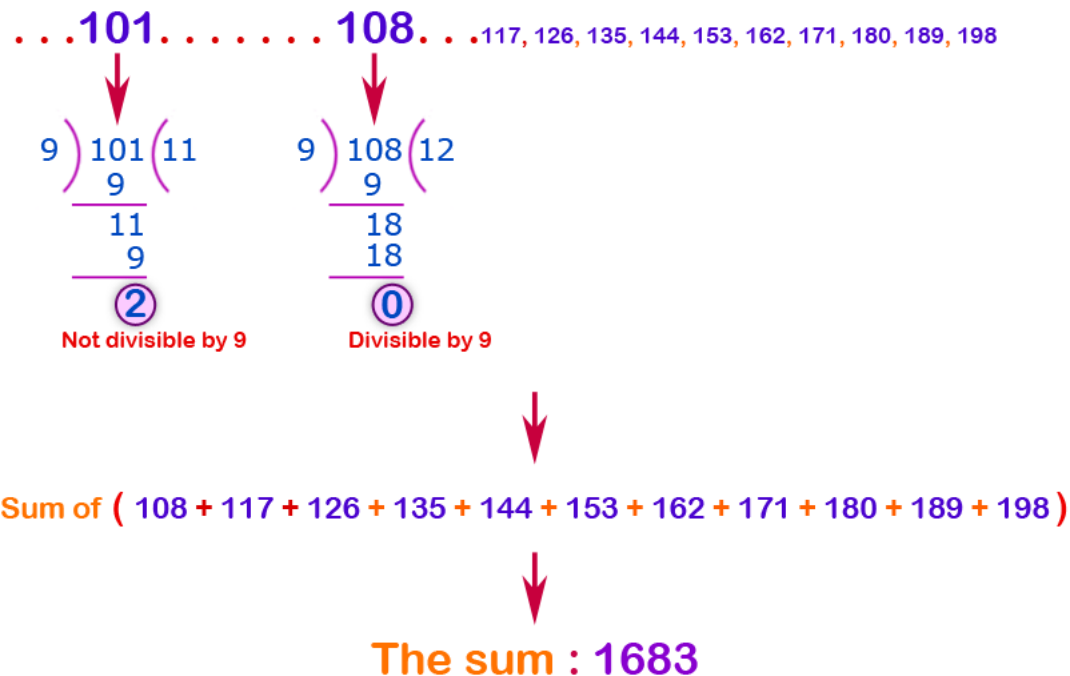
# Bài tập 7.41

- Write a program in C to check whether a number is a palindrome or not.



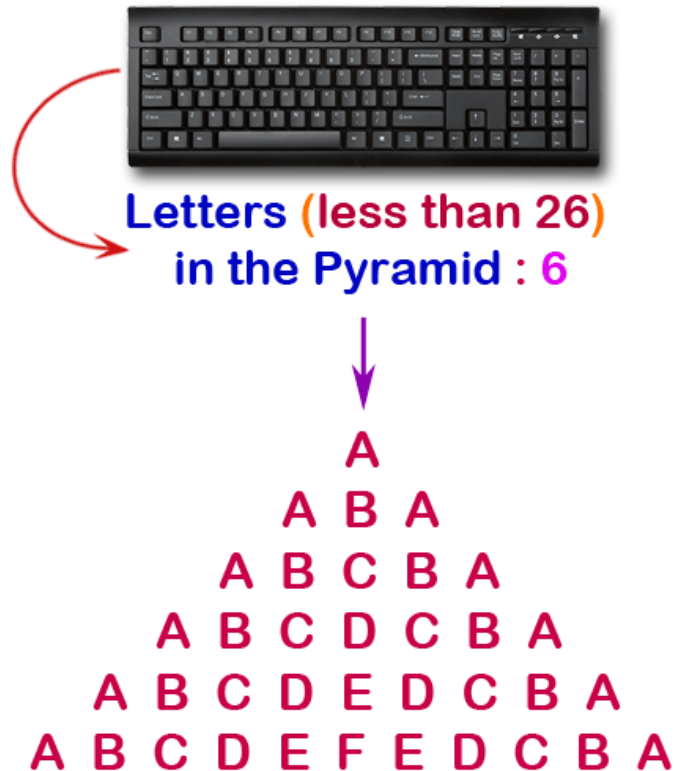
# Bài tập 7.42

- Write a program in C to find the number and sum of all integer between 100 and 200 which are divisible by 9.



# Bài tập 7.43

- Write a C Program to display the pattern like pyramid using the alphabet.



# Bài tập 7.44

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

## Decimal to Binary

2		47		
2		23	_____	1
2		11	_____	1
2		5	_____	1
2		2	_____	1
2		1	_____	0
		0	_____	1

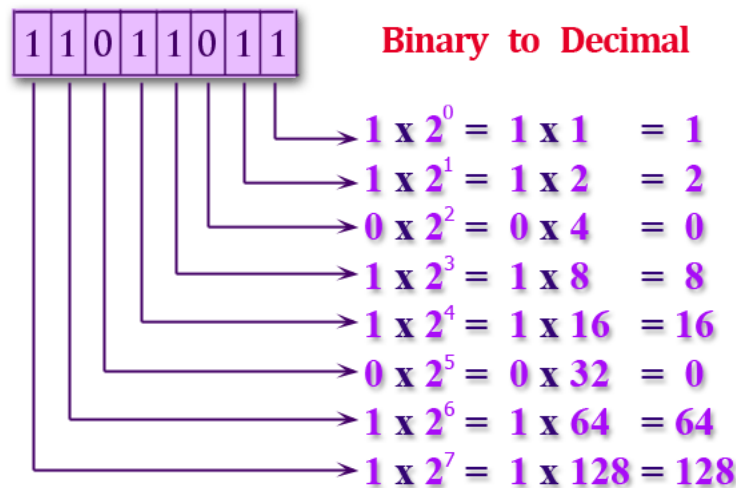
R  
e  
m  
a  
i  
n  
d  
e  
r

$$(47)_{10} = (101111)_2$$



# Bài tập 7.45

- Write a program in C to convert a binary number into a decimal number without using array, function and while loop.



$$1 + 2 + 8 + 16 + 64 + 128 = 219$$

$$(11011011)_2 = (219)_{10}$$

# Bài tập 7.46

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

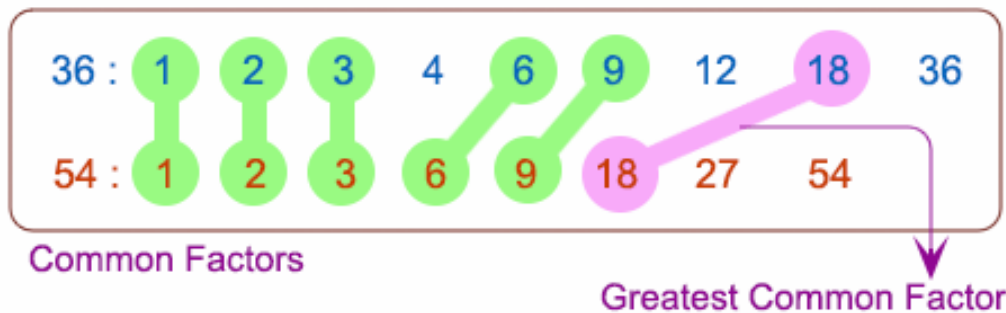
Determine the HCF of two numbers

List of Factors of 36 :

1 x 36, 2 x 18, 3 x 12, 4 x 9, 6 x 6

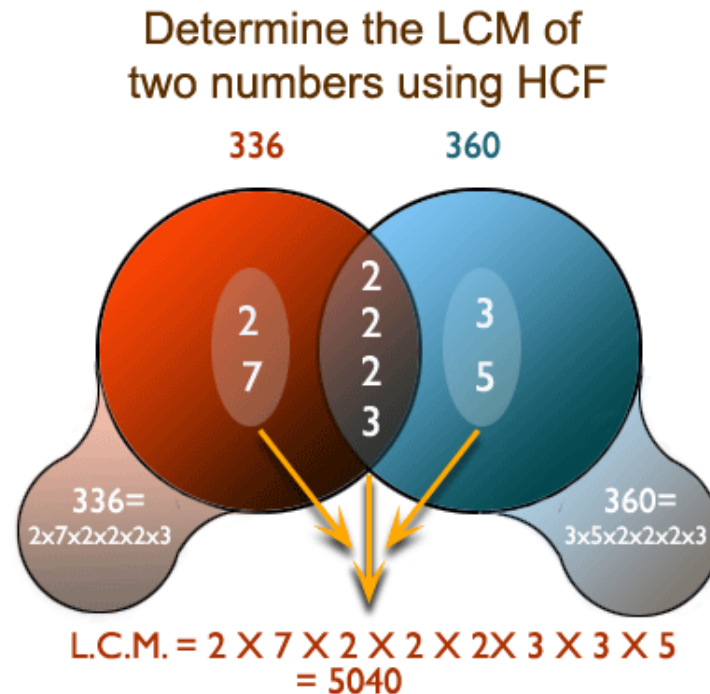
List of Factors of 54 :

1 x 54, 2 x 27, 3 x 18, 6 x 9



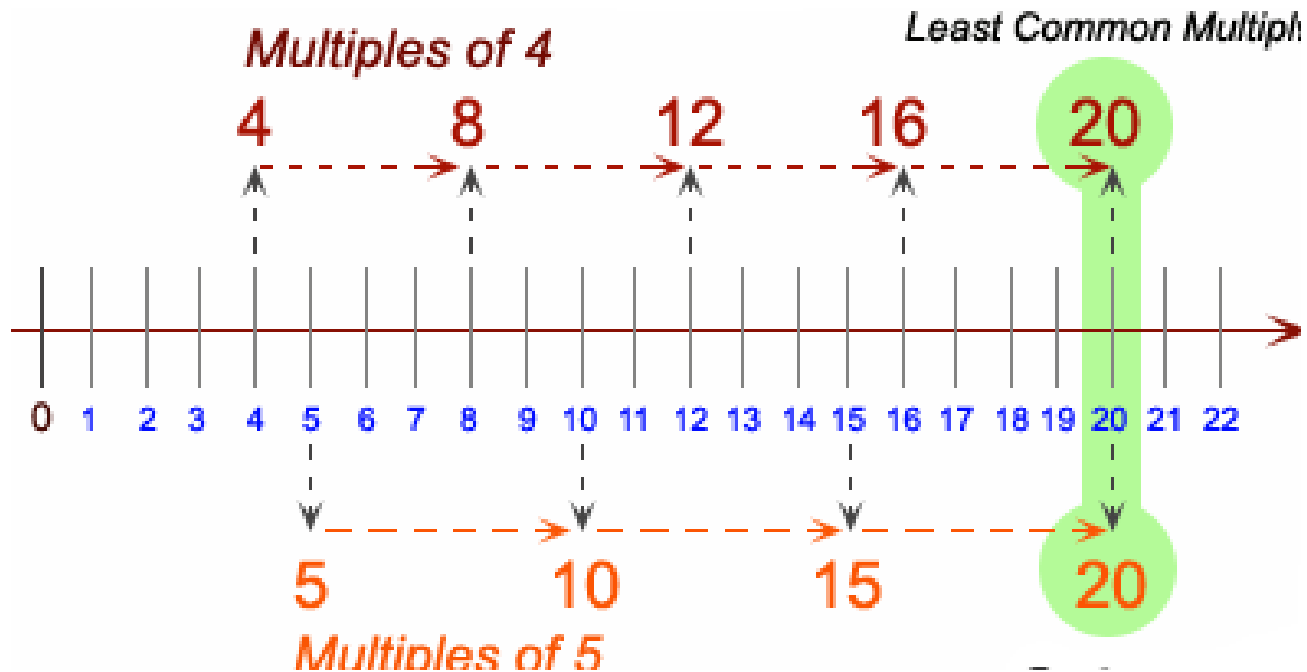
# Bài tập 7.47

- Write a program in C to find LCM of any two numbers using HCF.



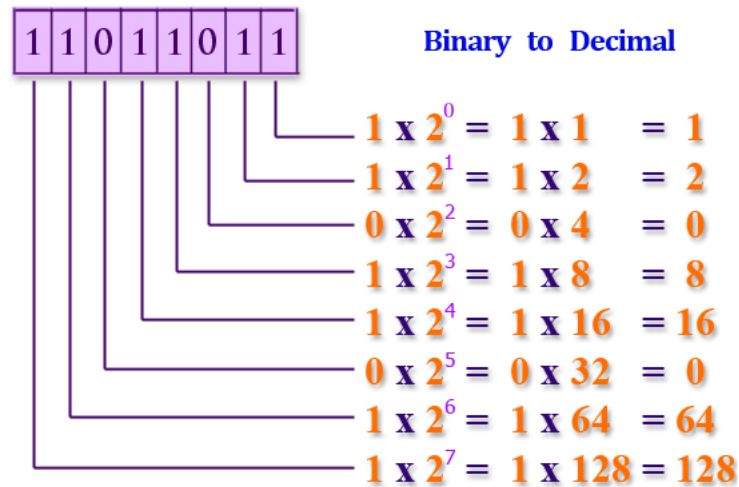
# Bài tập 7.48

- Write a program in C to find LCM of any two numbers.



# Bài tập 7.49

- Write a program in C to convert a binary number into a decimal number using math function.

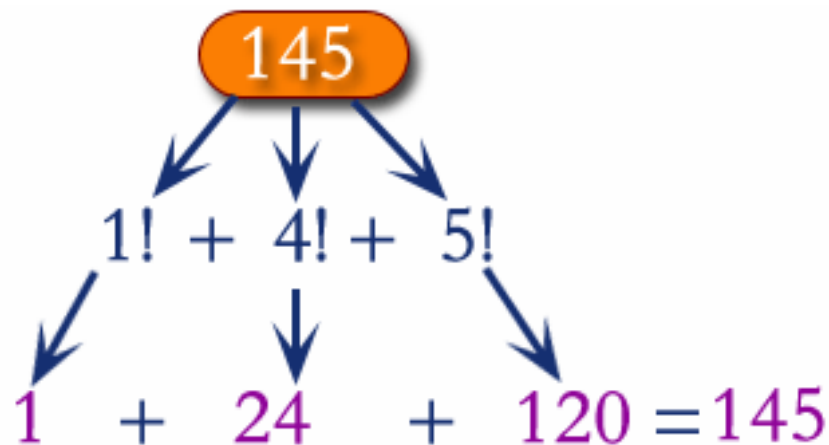


$$1 + 2 + 8 + 16 + 64 + 128 = 219$$

$$(11011011)_2 = (219)_{10}$$

# Bài tập 7.50

- Write a C program to check whether a number is a Strong Number or not.

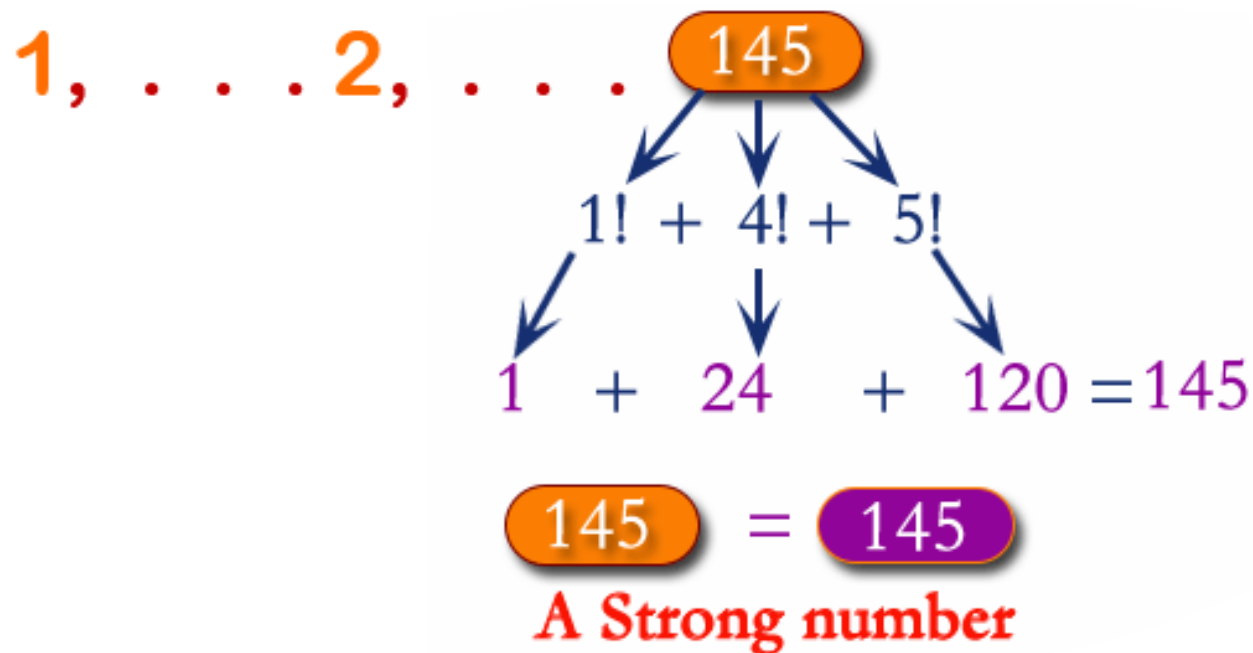


$$145 = 145$$

**A Strong number**

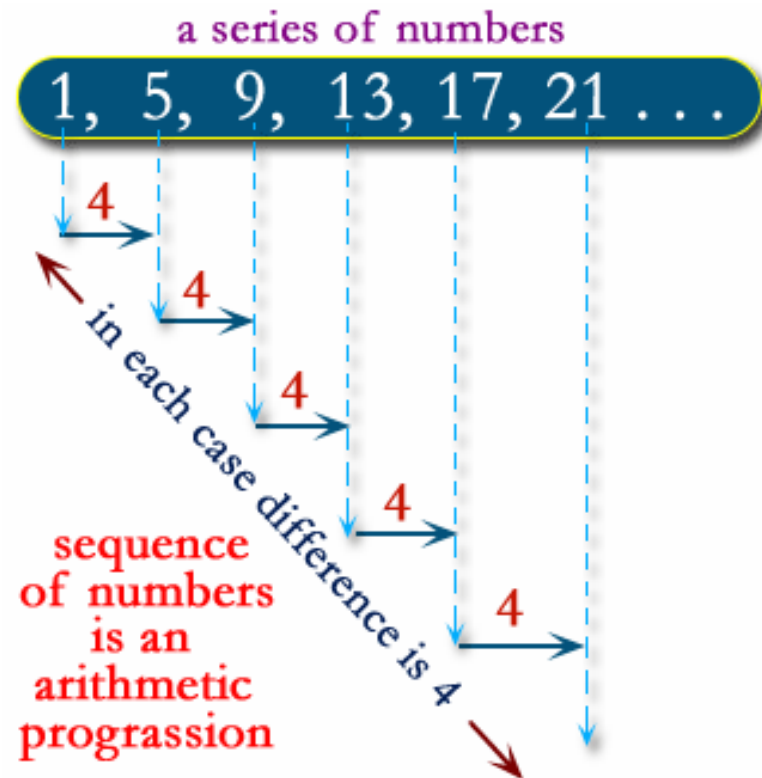
# Bài tập 7.51

- Write a C program to find Strong Numbers within a range of numbers.



# Bài tập 7.52

- Write a c program to find out the sum of an A.P. series.



The sum is :

$$1+5+9+13+17+21 = 66$$



# Bài tập 7.53

- Write a program in C to convert a Octal number to a Decimal number without using an array, function and while loop.

## Octal to Decimal

Octal	0	1	2	3	4	5	6	7
Decimal	0	1	2	3	4	5	6	7

Octal = 25

↓

2      5

$8^1$     $8^0$

$8 \times 2 = 16$     $1 \times 5 = 5$

$16 + 5$

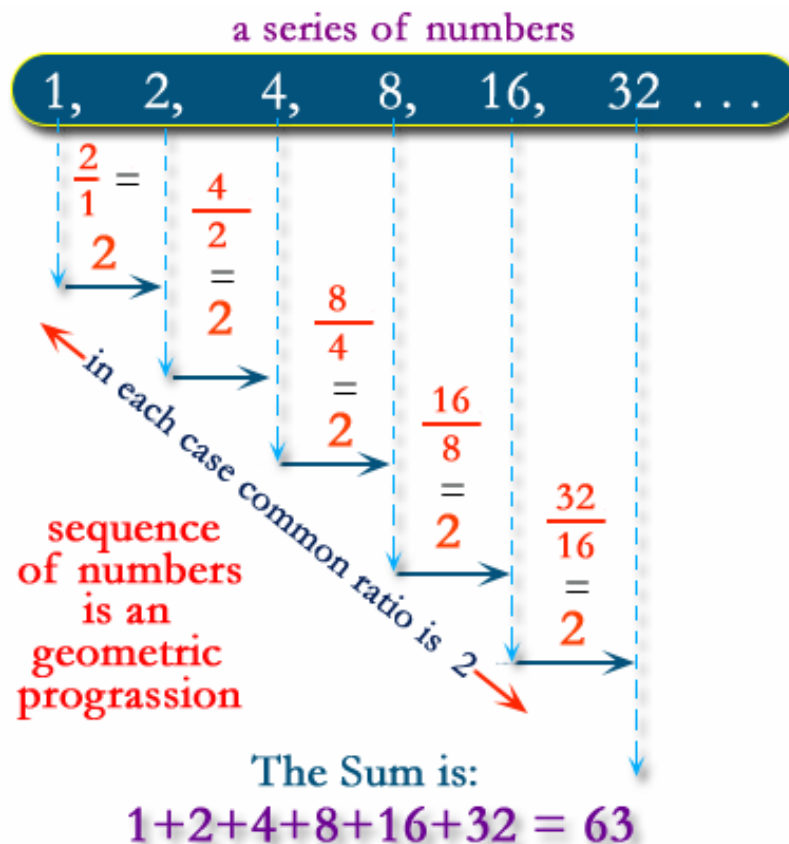
↓

21

$$(25)_8 = (21)_{10}$$

# Bài tập 7.54

- Write a program in c to find the Sum of GP series.



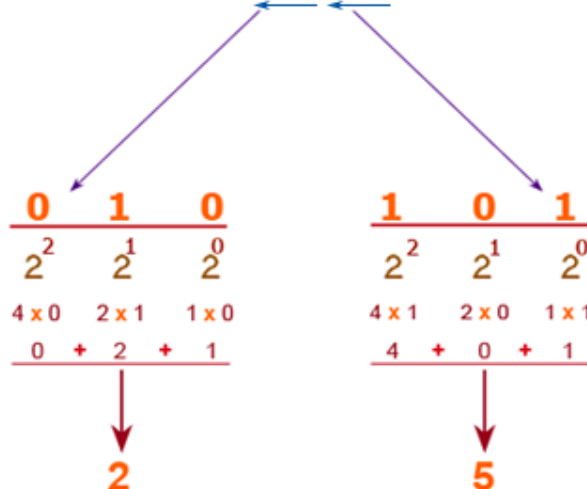
# Bài tập 7.55

- Write a program in C to convert a binary number to octal.

Binary to Octal

Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Binary Value = 010101



$$(10101)_2 = (25)_8$$

# Bài tập 7.56

- Write a program in C to convert an octal number into binary.

## Octal to Binary

Octal = 25

2

5

0 1 0  
 $2^2$   $2^1$   $2^0$

1 0 1  
 $2^2$   $2^1$   $2^0$

→ Binary Bits

$$(25)_8 = (010101)_2$$

# Bài tập 7.57

- Write a program in C to convert a decimal number to hexadecimal.

## Decimal to Hexadecimal

Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Decimal Value = 775

Integer Part	Quotient	Remainder	Remainder in Hexadecimal
775 / 16	48	7	7
48 / 16	3	0	0
3 / 16	0	3	3

Hexadecimal Value = 307

$$(775)_{10} = (307)_{16}$$

Decimal Value = 1256

Integer Part	Quotient	Remainder	Remainder in Hexadecimal
1256 / 16	78	8	8
78 / 16	4	14	E
4 / 16	0	4	4

Hexadecimal Value = 4E8

$$(1256)_{10} = (4E8)_{16}$$

# Bài tập 7.58

- Write a program in C to Check Whether a Number can be Express as Sum of Two Prime Numbers.

$16 = 1 + 15$  → Both are not prime

$16 = 2 + 14$  → 2 is prime but 14 is not

$16 = 3 + 13$  → Both are prime

$16 = 4 + 12$  → Both are not prime

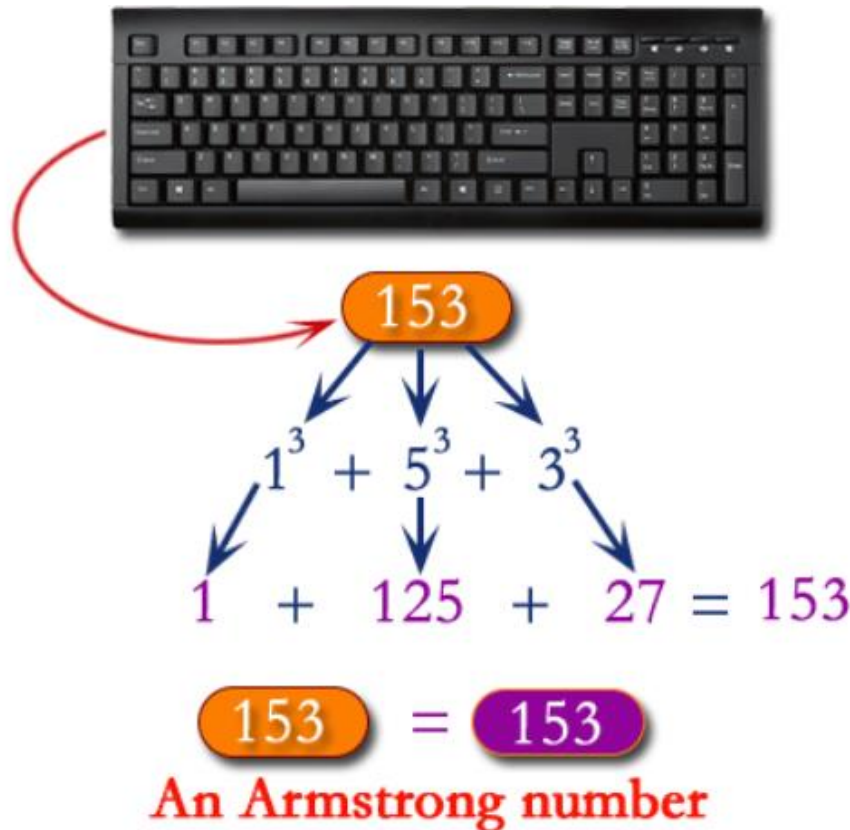
$16 = 5 + 11$  → Both are prime

$16 = 6 + 10$  → Both are not prime

$16 = 7 + 9$  → 7 is prime but 9 is not

# Bài tập 7.59

- Write a program in C to check Armstrong number of n digits.







25 YEARS ANNIVERSARY  
**SOICT**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you  
for your  
attentions!**



[soict.hust.edu.vn/](http://soict.hust.edu.vn/)



[fb.com/groups/soict](https://fb.com/groups/soict)

