

# C in One Shot

## Part - 5

COLLEGE  
WALLAH

# Functions in One Shot

COLLEGE  
WALLAH

# What and Why?

$$y = f(x)$$

$$y = f(x) = x^2 + 5$$



1) Repetition (to avoid) → loops

2) Readable

COLLEGE  
WALLAH

```
#include<stdio.h>
void greet(){
    ✓printf("Good Morning\n");
    ✓printf("How are you ?\n");
    return;
}
1 int main(){
2     greet();
    greet();
    greet();
    return 0;
}
```

External function

→ khatam

call

- Good Morning
- How are you ?
- Good Morning
- How are you ?
- Good Morning
- How are you ?
- 

DRY → do not repeat yourself

# Basic syntax

```
fun(){
```

```
// code
```

```
}
```

COLLEGE  
WALLAH

```
main() → starts from main()  
return; → khatam
```

COLLEGE  
WALLAH

**How functions work :** ek ke andar doosra,  
doosre ke andar teesra

COLLEGE  
WALLAH

## Kaam ki baate :

- 1) `main()` ek hi baar aata hai.
- 2) Starts with main
- 3) unlimited functions

COLLEGE  
WALLAH



**Return type** : Power function se samajte hai

COLLEGE  
WALLAH

Sum of 2 numbers

return type

```
int add(int a, int b) {
    return a + b;
}
```

arguments

3      4

a      b

main()

```
{ int a, b;
  scanf("%d %d", &a, &b);
  int sum = add(a, b);
}
```

3      4

a      b

COLLEGE  
WALLAH

# Library functions

```
printf ("Hello");
```

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

```
sqrt(49);
```

```
pow(2,5);     $\rightarrow$      $2^5$ 
```

COLLEGE  
WALLAH

# Function Prototype – ~~useless~~

COLLEGE  
WALLAH

# Ques : Combination and Permutation

$${}^nC_r = \frac{n!}{r! \times (n-r)!}$$

$$\begin{matrix} n=7 \\ r=3 \end{matrix} \rightarrow \frac{7!}{3! \cdot 4!}$$

```
int combination (int n, int r){
```

```
    //code
```

```
    {
```

$$\underbrace{(n-r)!}_{\substack{\downarrow \\ \text{number}}} = (n-r) \times (n-r-1) \times \dots$$

COLLEGE  
WALLAH

# Ques : Combination and Permutation

$$nCr = \frac{n \text{ fact}}{r \text{ fact}} \times \frac{n-r \text{ fact}}{1}$$

$$= \frac{n \text{ fact}}{r \text{ fact}} \times \frac{n-r \text{ fact}}{1}$$

$${}^nC_r = \frac{n!}{r! \times (n-r)!}$$

$${}^7C_3 = \frac{7!}{3! \times 4!} = \frac{7 \times \cancel{6} \times \cancel{5} \times \cancel{4} \times \cancel{3} \times \cancel{2} \times \cancel{1}}{\cancel{3} \times \cancel{2} \times \cancel{1} \times \cancel{4} \times \cancel{3} \times \cancel{2} \times \cancel{1}} = 35$$

$$0! = 1$$

$${}^n P_r = \frac{n!}{(n-r)!}$$

Homework:  ${}^n P_r \rightarrow \text{function}$

# Ques : Pascal triangle

	0	1	2	3	4	5	→ $r$
0	1						
1	1	1					
2	1	2	1				
3	1	3	3	1			
4	1	4	6	4	1		
5	1	5	10	10	5	1	
↑ $n$							

↓  
 $i$

	0	1	2	3	4	5	→ $r$
0	${}^0C_0$						
1	${}^1C_0$	${}^1C_1$					
2	${}^2C_0$	${}^2C_1$	${}^2C_2$				
3	${}^3C_0$	${}^3C_1$	${}^3C_2$	${}^3C_3$			
4	${}^4C_0$	${}^4C_1$	${}^4C_2$	${}^4C_3$	${}^4C_4$		
5							
↑ $n$							

←  $j$  →

↓

$iC_j$



# Ques : Pascal triangle OPTIMISED

Ultimate Method - 'Maths'

$$0 \quad {}^0C_0$$

$$1 \quad {}^1C_0 \quad {}^1C_1$$

$$2 \quad {}^2C_0 \quad {}^2C_1 \quad {}^2C_2$$

$$3 \quad {}^3C_0 \quad {}^3C_1 \quad {}^3C_2 \quad {}^3C_3$$

$$4 \quad {}^4C_0 \quad {}^4C_1 \quad {}^4C_2 \quad {}^4C_3 \quad {}^4C_4$$

$${}^nC_{r+1} = \frac{n!}{(r+1)! \times (n-r-1)!}$$

$$\Rightarrow {}^nC_{r+1} = \frac{n!}{(r+1) \cdot r! \cdot \frac{(n-r)!}{n-r}}$$

$$\Rightarrow {}^nC_{r+1} = \frac{n! \cdot (n-r)}{r! \cdot (n-r)! \cdot (r+1)}$$

$$(n-r)! = (n-r) \times (n-r-1)!$$

$$\Rightarrow \frac{(n-r)!}{n-r} = (n-r-1)!$$

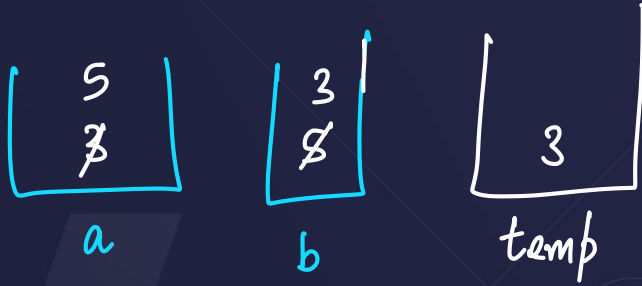
$$\Rightarrow {}^nC_{r+1} = {}^nC_r \times \left( \frac{n-r}{r+1} \right)$$

$${}^iC_{j+1} = {}^iC_j \times \frac{i-j}{j+1}$$

# Ques : Swap 2 numbers

```
int a = 3;
```

```
int b = 5;
```



```
int temp;
```

```
temp = a;
```

```
a = b;
```

```
b = temp;
```

**Ques** : Swap 2 numbers without using a third variable

$$\begin{array}{|l} \textcircled{3+5} \quad \cancel{3+5} \\ \hline 3+5 \\ \hline 8 \end{array}$$

$a$

$$\begin{array}{|l} \textcircled{3} \quad \cancel{5} \quad 5 \\ \hline 8 \end{array}$$

$b$

$$a = a + b;$$

$$b = a - b;$$

$$a = a - b;$$

COLLEGE  
WALLAH

# Scope of variable


 aukat  
 limit

swap( )  
 ↙      ↘  
 main()

{  
   int j = 4;  
   for(int i = 1; . . . ) {  
     3  
   }  
 }  
 3

COLLEGE  
WALLAH

# Formal parameters and Actual Parameters

```
void swap(int a, int b){
```

```
    int temp = a;
```

```
    a = b;
```

```
    b = temp;
```

```
    return;
```

```
}
```

```
int main(){
```

```
    a = 2;
```

```
    b = 4;
```

```
    swap(a, b);
}
```

a & b are formal

↓

2 & 4 are actual

# Pass by value & Pass by reference

```
void swap(int a, int b){
```

```
    int temp = a;
```

```
    a = b;
```

```
    b = temp;
```

```
    return;
```

```
}
```

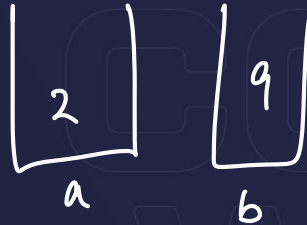
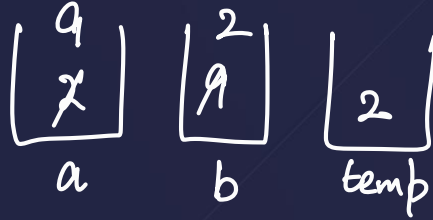
```
int main(){
```

```
    a = 2;
```

```
    b = 9;
```

```
    swap(a, b);
```

```
}
```



\* Notes

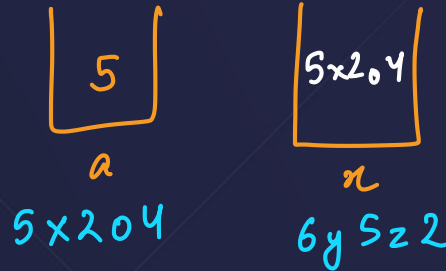
# \*Pointers

variable ka address store

```
int a = 5;
```

```
int* x = &a
```

```
*x = 7;
```



```
printf("%p", x); → 5x204
```

```
printf("%p", &x); → 6y5z2
```

```
printf("%d", *x); → 5
```

# \*Pointers

Swap 2 numbers 'pass by reference'

```
void swap(int a, int b){
```

```
    int temp = a;
```

```
    a = b;
```

```
    b = temp;
```

```
    return;
```

```
}
```

```
int main(){
```

```
    a = 2;
```

```
    b = 4;
```

```
    swap(&a, &b);
```

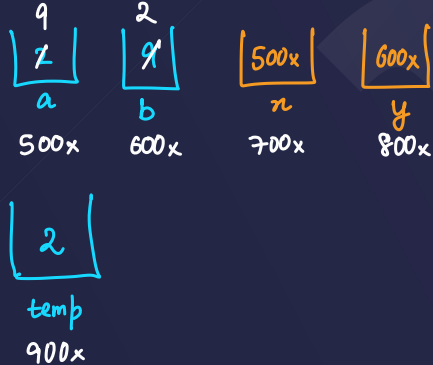
a & b are formal

↓

2 & 4 are actual



```
#include<stdio.h>
void swap(int* x, int* y){
    int temp;
    temp = *x; // temp = 2
    *x = *y; // a = 9
    *y = temp; // *y = 2 -> b = 2;
    return;
}
int main(){
    int a = 2;
    int b = 9;
    swap(&a,&b);
    printf("The value of a is %d\n",a);
    printf("The value of b is %d",b);
    return 0;
}
```



```
#include<stdio.h>
int main(){
    int a = 25;
    int* x = &a; //int*
    // VVIP -> *x = 7;
    int** y = &x; // int
    printf("%d\n",a);
    printf("%d\n",*x);
    printf("%d\n",**y);
    return 0;
}
```

25  
a

x300

x300  
x

x400

x400  
y

x500

\*x → 25 [%d]

\*y → x300 [%p]

\*\*y → 25 [%d]

1) Gate Wallah → C Programming

2) C++ → pointers

**Ques** : A positive integer is entered through the keyboard. Write a function to obtain the prime factors of this number.

(Hint : use separate function to check if a number is prime or not)

**Ques :** Write a function to compute the greatest common divisor of two given numbers

$a, b$        $GCD \rightarrow HCF$   
                                   $\downarrow$   
                                  highest common factor

$24 \rightarrow 1, 2, 3, 4, 6, 8, 12, 24$

$60 \rightarrow 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60$

`for (int i = 1 ; i < min(a,b) ; i++) {`

`}`

## State TRUE or FALSE :

- 1) The variables commonly used in C functions are available to all the functions in a program. *'False'*
- 2) To return the control back to the calling function we must use the keyword return. *True*
- 3) The same variable names can be used in different functions without any conflict. *True*

## State TRUE or FALSE :

- 4) Every called function must contain a return statement. *True*
- 5) A function may contain more than one return statements. *True → Only one should hit*
- 6) Each return statement in a function may return a different value. *True*

## State TRUE or FALSE :

7) A function can still be useful even if you don't pass any arguments to it and the function doesn't return any value back. *True*

8) Same names can be used for different functions without any conflict. *False*

**State TRUE or FALSE :**

9) A function may be called more than once from any other function *True*

10) It is necessary for a function to return some value.  
*No. False*



# HW : Print the factorials of first 'n' numbers

→ Using function

COLLEGE  
WALLAH

**HW** : Print first 'n' fibonacci numbers.

→ using functions

COLLEGE  
WALLAH