

Q

String

What is string?

A string is a sequence of characters treated as a group

(A word is a sequence of letter)

String is a group of character, (-)

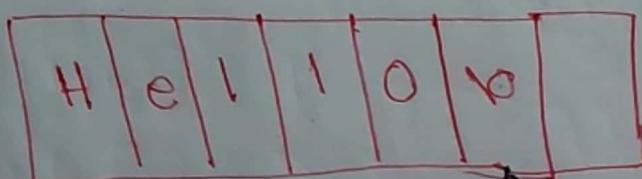
space allow.

Q for Representation in C

string কি তাকে → Data type এর

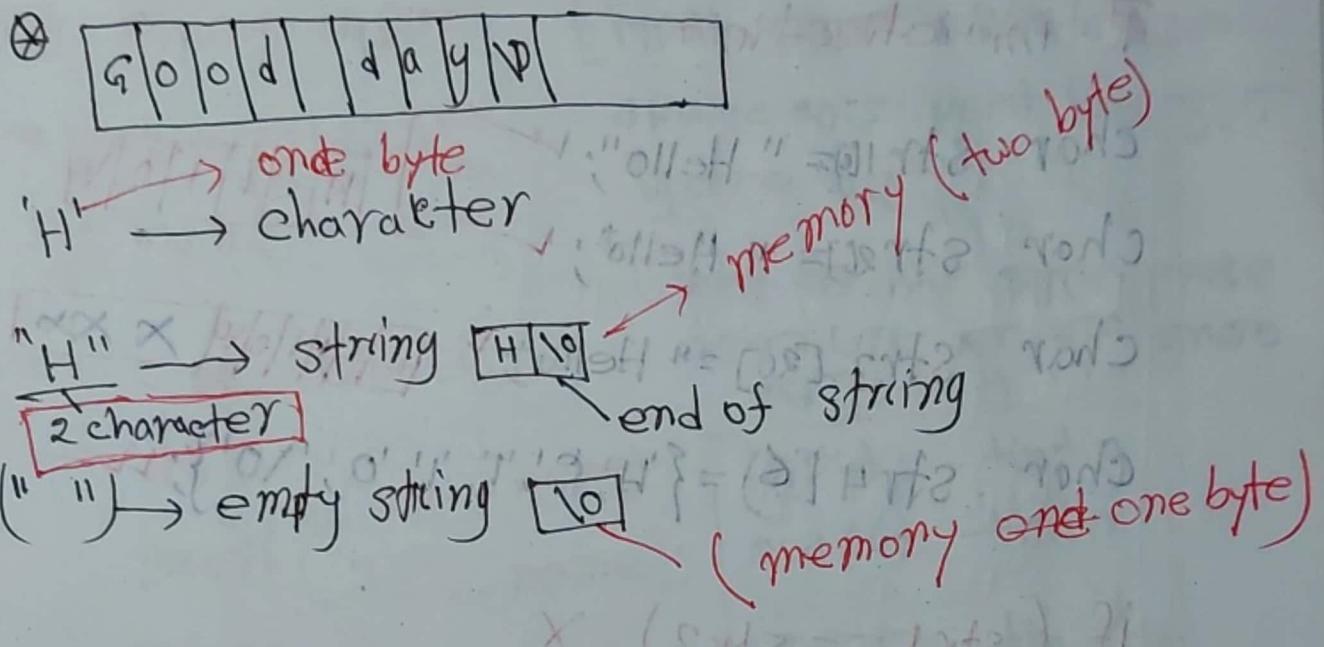


char temp [] = "Hello";



↑  
end of string

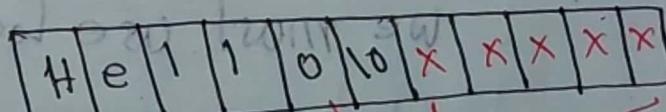
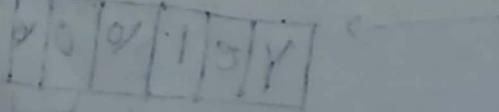
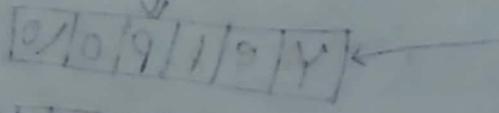
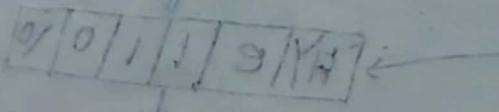
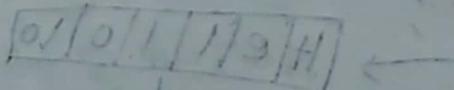
null / zero / end of string



⊗ char str[11] = "Hello";

char str[4] = "Hello";

invalid



string

part of array

## ④ initialization

char str1[] = "Hello"; ✓

char str2[] = "Hello"; ✓

char str3[20] = "Hello"; ✓

char str4[5] = {'H', 'e', 'l', 'l', 'o'}; ✓

if (str1 == str2) X

↓  
we must use loop

## ④ str1[0] = 'Y'; string to char

str1[0] = "Hello";

→ Hello

str1[0] = 'Y';

→ Yello

str1[3] = 'P';

→ Yeipolo

str1[3] = '\0'

→ Yelvoov

String

Array not  
String

## String input

```
#include <stdio.h>
int main()
{
    char name[10];
    scanf("%s", &name);
    printf("%s", name);
    return 0;
}
```

[input করেখাব এবং স্পেস অন্তে  
স্পেস থেকে print শুরু করে  
ignore করে]

[input করে স্ময় রাখে। sptake  
অন্তের টা শুরু হয়ে ignore  
করে]

input: tintin  
output: tintin

input: tim tim  
output: tim

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

```
int main()
```

```
{
```

```
char name[10];
```

```
gets(name);
```

```
printf("%s", name);
```

```
return 0;
```

```
}
```

input: Md Rakib

output: Md Rakib

input: method

gets(n)

scanf("%d", &n);

gets (n) gets (n); scanf("%s", name) [name]

| gets(n)  | puts scanf("%s", &n)  |
|--|---|
| ① प्रथम space / द्वितीय space<br>जरुर रखें। क्यों? | ① प्रथम space का लिया जाएगा।<br>द्वितीय group of word<br>प्राप्तिकरण का space नहीं।<br>लागत नहीं। |
| ② अन्त अहर तक input किया जाएगा।                    | ③   |
| ③  |   |

[printf एवं puts का नाम होता है (प्रिंट) वा (प्रिंट) विद्युत स्वरूप New line के बाहर

char name[10];

gets(name)

name[0] = 'H';

printf("

puts(name)

input. ki khobor

output. Hi khobor

```

char #include <stdio.h> #include <string.h>
int main()
{
    char name[10];
    gets(name);
    puts(name);
}

```

input: CS61111  
output: HSE1

name[0] = 'H';  
name[4] = '\0'; → end of string

puts str won't print

Q #include <stdio.h>

```
int main()
```

```
char str1[10] = "Hello";
```

```
printf("%s", str1);
```

```
puts(str1);
```

```
puts(str1);
```

```
printf("%s", namestr1);
```

puts str won't print  
first

HelloHello

Hello  
Hello

find the length

```
#include <stdio.h>
#include <string.h>
int main()
```

```
{ char str1[10] = "Hello";
```

```
int length = 0; // [0] = 'H' is start
```

```
int i = 0; // [0] = 'H' is start
```

```
while(str1[i] != '\0')
```

```
{
```

```
length++;
```

```
i++;
```

```
}
```

```
printf("The length is: %d\n", length);
```

```
}
```

or #include <string.h>

```
int length_1 = strlen(str1);
```

```
printf("The length is: %d\n", length_1);
```

# problem

Q1 write down a program that will print n-th letter in a sentence entered by a user.

n will be input to your program.

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

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

```
int main()
```

```
{
```

```
    char s[80];
```

```
    int n, length;
```

```
    printf ("Enter a sentence");
```

```
    gets (s);
```

```
    length = strlen(s);
```

```
    printf ("Total length: %.d\n", length);
```

```
    printf ("Which position?");
```

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

```
    if (n < length)
```

```
    {
```

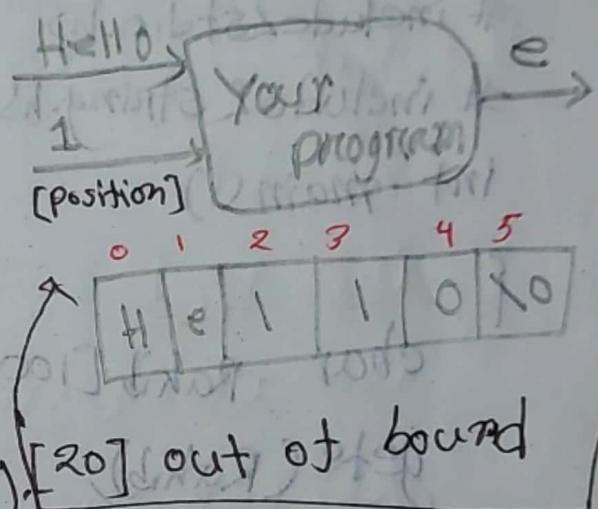
```
        printf ("The letter is: %c", s[n]);
```

```
    }
```

```
    else
```

```
        printf ("No letter at such position");
```

```
}
```



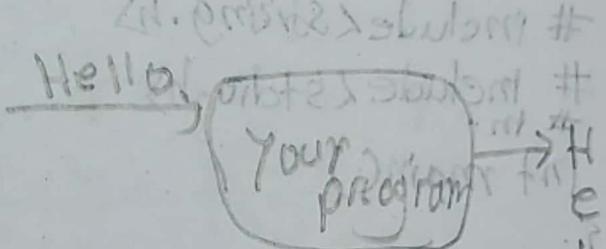
problem

solution

To write down a program that will print letter of a sentence in a vertical line.

Add delay as needed.

```
#include<stdio.h>
#include<string.h>
int main()
{
    char Rakib[100];
    gets(Rakib);
    int i;
    for(i=0; i<strlen(strRakib); i++)
    {
        printf("%c", Rakib[i]);
        sleep(500); // pause / delay
    }
    return 0;
}
```



Rakib[i] != '\0'

sleep always millisecond

sleep

sleep (500)

#include<windows.h>

problem

Write down a program that searches for a letter in a sentence. Both letter and sentence will be input to your program. print last position of the letter found in the sentence.

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

```
int main()
```

```
{
```

```
char R[100];
```

```
gets(R);
```

```
char x;
```

```
scanf("%c", &x);
```

```
int i=0, pos;
```

```
while(R[i]!='\0')
```

```
{
```

```
if (R[i]==x)
```

```
{
```

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

```
}
```

```
i++;
```

```
}
```

```
return 0;
```

```
}
```

input

Hello → ↗

output 1

input Hello

search: 1

output: 2, 3

input: Hello\_0\_!!

search: 1

output: 8

answer

→ ↗

input: Hello\_A!!

search: 1

output: 2

Lecture

Write down a program that prints how many words, letter, vowel and consonant exists in a sentence, the sentence will be inputs to your program.

```
#include<stdio.h>
#include<string.h>
int main()
{
    char R[100];
    gets(R);
    int length = strlen(R);
    int words=0, vowels=0, consonants=0, i;
    for(i=0; i<length; i++)
    {
        char t = tolower(R[i]);
        char temp = tolower(R[i+1]);
        if(t == ' ')
        {
            words++;
        }
        else
        {
            if((temp == 'a') || (temp == 'e') ||
               (temp == 'i') || (temp == 'o') || (temp == 'u'))
            {
                vowel++;
            }
        }
    }
    printf("Number of words = %d\n", words);
    printf("Number of vowels = %d\n", vowel);
    printf("Number of consonants = %d\n", consonants);
}
```

```
        else
    {
        consonant++;
    }
}

// end else
}

// for
```

~~return;~~

```
printf("Words : %d\n", words);
```

```
printf("Vowels : %d\n", vowels);
```

```
printf("Consonants: %d\n", consonants);
```

```
printf("Letters: %d\n", vowel + consonants);
```

~~return;~~

}

Inputs → love c programming

Outputs

words: 4

vowel: 7

consonant: 11

Letter: 17

## problem

write down a function that compares two string and returns 1 if they are same and returns 0 otherwise.

```
#include<stdio.h>
#include<string.h>
int main()
{
    char R1[100];
    char R2[100];
    int flag=1;
    gets(R1);
    gets(R2);
    if (strlen(R1) != strlen(R2))
    {
        printf flag=0;
    }
    else
    {
        for(int i=0; i < strlen(R1); i++)
        {
            if (str[R1][i] != str[R2][i])
            {
                flag = 0;
                break;
            }
        }
    }
}
```

```

if (flag == 1)
{
    printf ("Equal string");
}
else
{
    printf ("Not equal string");
}
return 0;
}

```

| input :- | output :- |
|----------|-----------|
| Hello    | not equal |
| Hello    | not equal |
| Hello    | equal     |
| Hello    | equal     |

~~problem~~

Write down a program that will take a word as input and will determine whether the word is palindrome or not.

```
#include <stdio.h>
#include <string.h>
int main()
{
    char R[100] = " ", rev[100] = "";
    gets(R); int flag = 1;
    for(int i=0; i<strlen(R); i++)
    {
        rev[i] = R[strlen(R)-1-i];
    }
    for(int i=0; i<strlen(R); i++)
    {
        if(R[i] != rev[i])
        {
            flag = 0; break;
        }
    }
    else
    {
        flag = 1;
    }
}
```

```

    }
    if (flag == 0)
    {
        printf ("Not palindrome");
    }
    else
    {
        printf ("palindrome");
    }
    return;
}

```

input:      rev      output  
 Hello     olleH      not palindrome  
 mam      mam      palindrome

another logic

```

for( int i=0; i<teststrlen(R); i++)
{
    if (str R[i] != R[ strlen(R)-1-i])
    {
        flag = 0; break;
    }
    else { flag = 1; }
}

```

## String Library Function

char str1[100] = "", str2[100] = "";

strcpy (\*str1, str2); ফার্মি করি কৈ

strncpy (\*str2, str1, 4); n জন্য রয়ে  
করি কৈ,

strncpy (\*str2, str1, 400); error

400 out of bound

strcat (\*str1, str2); || str1 = str1 + str2

strcat (\*str1, str1); || str1 = str1 + str1

strcat ("(str1, "END"); || str1 = str1 + "END"

infinity loop

strcmp (s, t) কমপ্লিক কথা, আচি t বড়

হব তখনে কি return রয়ে নেগেটিভ value

আচি s হব হব তখন return পোজিটিভ value

[ compare করে ASCII value কৈ ]

সমস্যা হল return করবে (0)

strcmpi (s, t) [ str1 এবং str2 same case  
compare কৈ ]

~~problem~~

Homework

palindrome check

```
#include<stdio.h>
#include<string.h>
int main()
{
    char s[100] = "madam";
    char t[100];
    gets(s);
    strcp(t,s);
    strrev(t);
    if (strcmpi(s,t) == 0)
    {
        printf("%s", "is a plindrom", s);
    }
    else
    {
        printf("%s", "is not a plindrom", s);
    }
    return 0;
}
```

problem

## word search

write down a program that searches for a word and in a sentence. Both word and sentence will be input to your program.  
print first position of the word found in the sentence.

```
#include<stdio.h>  
#include<string.h>  
int main()  
{  
    char s[100], t[100];  
    int i, p, j;  
    printf("Enter a sentence : ");  
    gets(s);  
    printf("Which word ? ");  
    gets(t);  
    p = -1;  
    for(i = 0; i < strlen(s); i++)
```

```

if (s[i] == t[0])
{
    for (j = 1; j < strlen(t); j++)
    {
        if (s[i+j] != t[j])
        {
            break;
        }
    }
    if (j == strlen(t))
    {
        p = i
        break;
    }
}
// Outer loop end
if (p == -1)
    printf (" sorry not found");
else
    printf (" Found at position: %d", p);
return 0;
}

```

## Function

Function, main code এর দ্বারা দেখানো মডেল  
করা করা।

- ① input
- ② output

function call করা কিম করবে কিম return  
যথেত, এ হল function name, (parameter list)

## User define function

- ① readability improve
  - ② modification easier
  - ③ Troubleshooting / Debug easier
- } advantage

## By

- ① Function definition
- ② parameters / argument →
  - ① actual
  - ② formal
- ③ Function call
- ④ prototype
- ⑤ Local / Global variable
- ⑥ return value

## Function definition

The diagram illustrates the function call mechanism between the `main()` and `add()` functions.

**Function Prototype:** `int add ( int , int );`

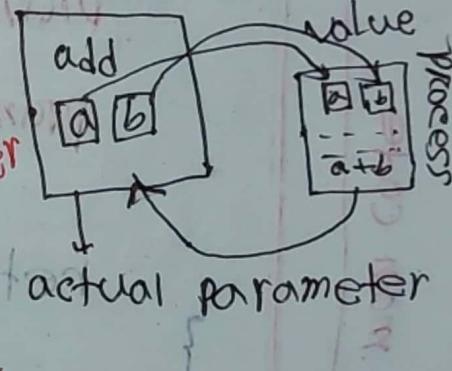
- Function name:** `add`
- return type:** `int`
- formal parameter:** `int n, int y`
- actual parameter:** `a, b`
- call:** `printf( "%d", add ( a+b ) );`

**Function Definition:** `int add ( int x, int y ) {`

- Function name:** `add`
- return type:** `int`
- formal parameter:** `x, y`
- actual parameter:** `a, b`
- process:** `int d = (a+b);`
- return:** `return d;`

**Annotations:**

- Function name:** `add` (highlighted in red)
- return user func:** `int` (highlighted in red)
- actual parameters:** `a, b` (highlighted in red)
- call করা বলে:** `printf( "%d", add ( a+b ) );` (highlighted in red)
- actual parameter:** `a, b` (highlighted in red)
- return type:** `int` (highlighted in red)
- function name:** `add` (highlighted in red)
- formal parameter:** `x, y` (highlighted in red)
- actual parameter:** `a, b` (highlighted in red)
- process দ্বারা:** `int d = (a+b);` (highlighted in red)
- return:** `return d;` (highlighted in red)
- global main এর সোজে আছে**
- সোজে use করাও আছে।**
- কোড়ো**



global main गया बाइबेल थाए  
— अमेरिका या अंग्रेजी भाषा में

Local main cost factors  
ব্যয়।

global ~~g~~- memory

main ସବୁ ମାରିବେ ଥାଏ ୨୦୯୫  
କାହାର ମାରି ଯାଏବା କାହାର ମା

part of function

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

```
int max(int, int, int); // Function prototype
```

```
int main()
```

```
{ printf("%d", max(10, 20, 30)); // 30
```

```
printf("%d", max(20, 1000, 30)); // 1000
```

```
return 0;
```

```
int max(int n, int y, int z)
```

```
{ int temp;
```

```
if (n > y && n > z)
```

```
temp = n;
```

```
else if (y > n && y > z)
```

```
temp = y;
```

```
else
```

```
temp = z;
```

```
return temp;
```

main code

for loop  
function name  
max কিৰি নথি  
কাজ কৰে এখন  
ৰেখা, use কৰ  
ৰেখা-ৰেখা কৰ  
তাৰে -& call কৰ  
ৰেখা  
ৰেখা-ৰেখা  
main কৰ বৰো  
printf কৰ বৰো

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

```
void man( int, int, int); // prototype
```

```
int main()
```

```
{
```

```
    man( 10, 20, 100);
```

```
    man( 200, 500, 100);
```

```
    return 0;
```

```
}
```

```
void man( int n, int y, int z)
```

```
{
```

```
    int temp;
```

```
    if (n > y && n > z)
```

```
        temp = n;
```

```
    else if (y > n && y > z)
```

```
        temp = y;
```

```
    else
```

```
        temp = z;
```

function call  
maximum of  
10, 20, 100 is 100

function call  
maximum of  
200, 500, 100 is 500

maximum function  
maximum of  
n, y, z is n

function call  
maximum of  
n, y, z is y

function call  
maximum of  
n, y, z is z

The maximum of %d, %d, %d is %d  
(n, y, z, temp);

```
}
```

$\lceil \text{math.h} \rceil$  always return floating number.

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

```
int power(int int); // prototype
```

```
int main()
```

```
{ int a, b;
```

```
scanf("%d %d", &a, &b); // function call
```

```
printf("%d\n", power(a, b)); // R5
```

```
return;
```

```
}
```

```
int power(int n, int y)
```

```
{ int product = 1, i;
```

```
for (i=1; i<=y; i++) // for loop to
```

```
{ product = product * n; // product = product * n (i.e.)
```

```
}
```

```
return product;
```

```
}
```

## String Length

```
#include <stdio.h>
int string_length(char []);
int main()
{
    char s[100];
    gets(s);
    int len = string_length(s);
    printf("%d", len);
    return 0;
}
return 0;

int string_length(char str[])
{
    int i=0;
    while(str[i] != '\0')
    {
        i++;
    }
    return i;
}
```

# add

#include <stdio.h>

void add(); // prototype

int main()

{

add();

return 0;

}

void add()

{

int n, y;

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

int sum = n + y;

printf("The sum of %d and %d is %d\n", n, y, sum);

}

~~problem~~

213

10) 213(21

rev=3

write down a function that will take an integer as parameter and will return the reverse of the number

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

int reverse ( int

```
int main()
```

{

int.  $\mu y$ ;  $n = 12345678 = N$  tri

```
y = reverse(n);  
printf ("Reversing: %.d\n", y);
```

return 0;

}

int reverse ( int num )

1

int r, \*sunrev; i=1; r = 0;

while (~~num! = 0~~)

8

$r = num \cdot 10^{\circ}$

$\text{num} = \text{num} / 10;$

rev = rev \* 10 + r;

~~return rev;~~

3

3

~~problem~~

19818(01) 818

# write down a function that will take an integer as parameter and will return number of digits on it.

#include<stdio.h>

```
int Howmanydigit ( int );  
int main()  
{  
    int n = 3456;  
    printf ("%d\n", Howmanydigit (n));  
    return 0;  
}  
  
int Howmanydigit ( int n )  
{  
    int r, c=0;  
    while (n>0)  
    {  
        r = n % 10;  
        n = n / 10;  
        c++;  
    }  
    return c;  
}
```

problem  
white down a function that will take two integers  
 $n$  and  $y$  as parameters and will return  $\text{GCD}(n, y)$ .

```
#include <stdio.h>
int GCD (int, int);
int main ()
{
    int n = 3, y = 17;
    printf ("%d\n", GCD (n, y));
    return 0;
}

int GCD (int n, int y)
{
    if (n > y)
    {
        man = n, min = y;
    }
    else
    {
        man = y, min = n;
    }
    manur of min;
```

$$\begin{array}{r} 3, 17 \\ \boxed{3, 17} \end{array}$$
 GCD

$$\begin{array}{r} 3) 17(85 \\ 15 \\ \hline 2) 32(1 \\ 2 \\ \hline 0 \end{array}$$
  $\boxed{\text{GCD}}$

problem  
V.V.I

n and then tell the next prime or  
last prime!

write down a function that will take an integer as parameters and will return 1 if the integer is a prime number and return 0 if otherwise

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

```
int prime (int);
```

```
int main()
```

```
{ int y = 17 n, i;
```

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

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

```
{ printf ("%d", prime(i)); }
```

```
}
```

```
int prime (int n)
```

```
{ int flag = 1, i;
```

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

```
{ if (n % i == 0)
```

```
{ flag = 0;
```

```
} } break;
```

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

```
{ if (prime(i) == 1)
```

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

```
}
```

```
else
```

```
return flag;
```

## Recursion

### recursion

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

```
Void print()
```

```
{ int main
```

```
{
```

```
print();
```

```
}
```

```
Void print()
```

```
{
```

```
printf("Have a good day");
```

```
print();
```

```
{
```

recursion ~~function~~ ~~function~~  
~~function~~ function ~~function~~  
— function ~~call~~  
— kind of  
infinity loop

```
}
```

```
{
```

```
((i==n) return return)
```

```
{
```

Q n नियम द्वारा print करें

#include <stdio.h>

void print\_recursion (int);

int main()

{ cout << "Enter value"; }

print\_recursion (x);

cout << endl;

return 0;

}

(1) Have a good day!!!

(2) Have a good day!!!

(3) Have a good day!!!

Void print\_recursion (n: int n)

{

printf("Have a good day !!!");

if (n == 1)

{ return; }

}

else

{ print\_recursion (n-1); }

}

}

loop का अंत तक / base case

return का main से आवश्यक

```

#include <stdio.h>
void callme(); // (1 - n) fi
int main()
{
    callme();
    return 0;
}
void callme()
{
    callme(); // (1 - n) sm/1/05
    printf("Hello !!\n");
    printf("Hello !!\n"); } infinity) fi
    callme(); // (1 - n) sm/1/05
    printf("Hello !! %d \n"); } Hello 10 !
    retif(n = 1) return ; } - - -
    callme(n - 1); } Hello 1 !!

```

in output নির্মাণ  
and মূল পদ্ধতি হচ্ছে

```
if (n == 1) return;
printf ("Hello!! %d\n", n);
callme (n-1);
```

(1-iteration) Hello !! 10  
Hello !! 10  
Hello !! 2

```
printf ("Hello !! %d\n", n);
callme (n-1);
if (n == 1) return;
callme (n-1);
if (n == 1) return;
printf ("Hello !! %d\n", n);
```

(2-iteration) Hello !!  
infinity

```
if (n == 1) return;
callme (n-1);
printf ("Hello !! %d\n", n);
```

no output

(3-iteration) Hello !! 2  
Hello !! 2  
Hello !! 5

## Recursion tree

callme(5)

|

callme(4)

Hello !! 5

callme(9)

Hello !! 4

(8) sm1100

Hello !! 3

(9) tri sm1100 b10v

callme(1)

Hello !! 2

break

What is Recursion?

When a function calls itself, the phenomenon is called Recursion.

For successful recursion the self calling must be stopped.

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

```
void callme (int);
```

```
int main ( )
```

```
{
```

```
    callme (2);
```

```
}
```

```
{
```

```
    void callme (int n)
```

```
{
```

```
    if (n==0) return n;
```

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

```
    callme(n-1);
```

```
    printf ("Biday prithibi%d\n");
```

```
}
```

Hello 2  
Biday #1

Hello 2

Hello 1

B.P 1

B.P 2

The

```
callme(2)
```

Hello 2

```
callme(1)
```

Biday prithibi 2

Hello 1

```
callme(0)
```

Biday prithibi 1

problem

factorial

#include <stdio.h>

int fact (int);

{

int n;

scanf ("%d", &n);

printf ("! = %d", n, fact (n));

}

int fact (int n)

if (n <= 1) return n;

return n \* fact (n - 1);

}

$$0 = \{ 1 \} \quad \{ 1 - 1 \} = 1$$

<defining solution>

(n, fact (n)) w.r.t. n

(n - 1, fact (n - 1)) w.r.t. n

(n - 2, fact (n - 2)) w.r.t. n

(n - 3, fact (n - 3)) w.r.t. n

(n - 4, fact (n - 4)) w.r.t. n

(n - 5, fact (n - 5)) w.r.t. n

(n - 6, fact (n - 6)) w.r.t. n

(n - 7, fact (n - 7)) w.r.t. n

(n - 8, fact (n - 8)) w.r.t. n

(n - 9, fact (n - 9)) w.r.t. n

(n - 10, fact (n - 10)) w.r.t. n

(n - 11, fact (n - 11)) w.r.t. n

(n - 12, fact (n - 12)) w.r.t. n

(n - 13, fact (n - 13)) w.r.t. n

(n - 14, fact (n - 14)) w.r.t. n

(n - 15, fact (n - 15)) w.r.t. n

(n - 16, fact (n - 16)) w.r.t. n

(n - 17, fact (n - 17)) w.r.t. n

(n - 18, fact (n - 18)) w.r.t. n

(n - 19, fact (n - 19)) w.r.t. n

(n - 20, fact (n - 20)) w.r.t. n

(n - 21, fact (n - 21)) w.r.t. n

(n - 22, fact (n - 22)) w.r.t. n

(n - 23, fact (n - 23)) w.r.t. n

(n - 24, fact (n - 24)) w.r.t. n

(n - 25, fact (n - 25)) w.r.t. n

(n - 26, fact (n - 26)) w.r.t. n

(n - 27, fact (n - 27)) w.r.t. n

(n - 28, fact (n - 28)) w.r.t. n

(n - 29, fact (n - 29)) w.r.t. n

(n - 30, fact (n - 30)) w.r.t. n

(n - 31, fact (n - 31)) w.r.t. n

(n - 32, fact (n - 32)) w.r.t. n

(n - 33, fact (n - 33)) w.r.t. n

(n - 34, fact (n - 34)) w.r.t. n

(n - 35, fact (n - 35)) w.r.t. n

(n - 36, fact (n - 36)) w.r.t. n

(n - 37, fact (n - 37)) w.r.t. n

(n - 38, fact (n - 38)) w.r.t. n

(n - 39, fact (n - 39)) w.r.t. n

(n - 40, fact (n - 40)) w.r.t. n

(n - 41, fact (n - 41)) w.r.t. n

(n - 42, fact (n - 42)) w.r.t. n

(n - 43, fact (n - 43)) w.r.t. n

(n - 44, fact (n - 44)) w.r.t. n

(n - 45, fact (n - 45)) w.r.t. n

(n - 46, fact (n - 46)) w.r.t. n

(n - 47, fact (n - 47)) w.r.t. n

(n - 48, fact (n - 48)) w.r.t. n

(n - 49, fact (n - 49)) w.r.t. n

(n - 50, fact (n - 50)) w.r.t. n

(n - 51, fact (n - 51)) w.r.t. n

(n - 52, fact (n - 52)) w.r.t. n

(n - 53, fact (n - 53)) w.r.t. n

(n - 54, fact (n - 54)) w.r.t. n

(n - 55, fact (n - 55)) w.r.t. n

(n - 56, fact (n - 56)) w.r.t. n

(n - 57, fact (n - 57)) w.r.t. n

(n - 58, fact (n - 58)) w.r.t. n

(n - 59, fact (n - 59)) w.r.t. n

(n - 60, fact (n - 60)) w.r.t. n

(n - 61, fact (n - 61)) w.r.t. n

(n - 62, fact (n - 62)) w.r.t. n

(n - 63, fact (n - 63)) w.r.t. n

(n - 64, fact (n - 64)) w.r.t. n

(n - 65, fact (n - 65)) w.r.t. n

(n - 66, fact (n - 66)) w.r.t. n

(n - 67, fact (n - 67)) w.r.t. n

(n - 68, fact (n - 68)) w.r.t. n

(n - 69, fact (n - 69)) w.r.t. n

(n - 70, fact (n - 70)) w.r.t. n

(n - 71, fact (n - 71)) w.r.t. n

(n - 72, fact (n - 72)) w.r.t. n

(n - 73, fact (n - 73)) w.r.t. n

(n - 74, fact (n - 74)) w.r.t. n

(n - 75, fact (n - 75)) w.r.t. n

(n - 76, fact (n - 76)) w.r.t. n

(n - 77, fact (n - 77)) w.r.t. n

(n - 78, fact (n - 78)) w.r.t. n

(n - 79, fact (n - 79)) w.r.t. n

(n - 80, fact (n - 80)) w.r.t. n

(n - 81, fact (n - 81)) w.r.t. n

(n - 82, fact (n - 82)) w.r.t. n

(n - 83, fact (n - 83)) w.r.t. n

(n - 84, fact (n - 84)) w.r.t. n

(n - 85, fact (n - 85)) w.r.t. n

(n - 86, fact (n - 86)) w.r.t. n

(n - 87, fact (n - 87)) w.r.t. n

(n - 88, fact (n - 88)) w.r.t. n

(n - 89, fact (n - 89)) w.r.t. n

(n - 90, fact (n - 90)) w.r.t. n

(n - 91, fact (n - 91)) w.r.t. n

(n - 92, fact (n - 92)) w.r.t. n

(n - 93, fact (n - 93)) w.r.t. n

(n - 94, fact (n - 94)) w.r.t. n

(n - 95, fact (n - 95)) w.r.t. n

(n - 96, fact (n - 96)) w.r.t. n

(n - 97, fact (n - 97)) w.r.t. n

(n - 98, fact (n - 98)) w.r.t. n

(n - 99, fact (n - 99)) w.r.t. n

(n - 100, fact (n - 100)) w.r.t. n

(n - 101, fact (n - 101)) w.r.t. n

(n - 102, fact (n - 102)) w.r.t. n

(n - 103, fact (n - 103)) w.r.t. n

(n - 104, fact (n - 104)) w.r.t. n

(n - 105, fact (n - 105)) w.r.t. n

(n - 106, fact (n - 106)) w.r.t. n

(n - 107, fact (n - 107)) w.r.t. n

(n - 108, fact (n - 108)) w.r.t. n

(n - 109, fact (n - 109)) w.r.t. n

(n - 110, fact (n - 110)) w.r.t. n

(n - 111, fact (n - 111)) w.r.t. n

(n - 112, fact (n - 112)) w.r.t. n

(n - 113, fact (n - 113)) w.r.t. n

(n - 114, fact (n - 114)) w.r.t. n

(n - 115, fact (n - 115)) w.r.t. n

(n - 116, fact (n - 116)) w.r.t. n

(n - 117, fact (n - 117)) w.r.t. n

(n - 118, fact (n - 118)) w.r.t. n

(n - 119, fact (n - 119)) w.r.t. n

(n - 120, fact (n - 120)) w.r.t. n

(n - 121, fact (n - 121)) w.r.t. n

(n - 122, fact (n - 122)) w.r.t. n

(n - 123, fact (n - 123)) w.r.t. n

(n - 124, fact (n - 124)) w.r.t. n

(n - 125, fact (n - 125)) w.r.t. n

(n - 126, fact (n - 126)) w.r.t. n

(n - 127, fact (n - 127)) w.r.t. n

(n - 128, fact (n - 128)) w.r.t. n

(n - 129, fact (n - 129)) w.r.t. n

(n - 130, fact (n - 130)) w.r.t. n

(n - 131, fact (n - 131)) w.r.t. n

(n - 132, fact (n - 132)) w.r.t. n

(n - 133, fact (n - 133)) w.r.t. n

(n - 134, fact (n - 134)) w.r.t. n

(n - 135, fact (n - 135)) w.r.t. n

(n - 136, fact (n - 136)) w.r.t. n

(n - 137, fact (n - 137)) w.r.t. n

(n - 138, fact (n - 138)) w.r.t. n

(n - 139, fact (n - 139)) w.r.t. n

(n - 140, fact (n - 140)) w.r.t. n

(n - 141, fact (n - 141)) w.r.t. n

(n - 142, fact (n - 142)) w.r.t. n

(n - 143, fact (n - 143)) w.r.t. n

(n - 144, fact (n - 144)) w.r.t. n

(n - 145, fact (n - 145)) w.r.t. n

(n - 146, fact (n - 146)) w.r.t. n

(n - 147, fact (n - 147)) w.r.t. n

(n - 148, fact (n - 148)) w.r.t. n

(n - 149, fact (n - 149)) w.r.t. n

(n - 150, fact (n - 150)) w.r.t. n

(n - 151, fact (n - 151)) w.r.t. n

(n - 152, fact (n - 152)) w.r.t. n

(n - 153, fact (n - 153)) w.r.t. n

(n - 154, fact (n - 154)) w.r.t. n

(n - 155, fact (n - 155)) w.r.t. n

(n - 156, fact (n - 156)) w.r.t. n

(n - 157, fact (n - 157)) w.r.t. n

(n - 158, fact (n - 158)) w.r.t. n

(n - 159, fact (n - 159)) w.r.t. n

(n - 160, fact (n - 160)) w.r.t. n

(n - 161, fact (n - 161)) w.r.t. n

(n - 162, fact (n - 162)) w.r.t. n

(n - 163, fact (n - 163)) w.r.t. n

(n - 164, fact (n - 164)) w.r.t. n

(n - 165, fact (n - 165)) w.r.t. n

(n - 166, fact (n - 166)) w.r.t. n

(n - 167, fact (n - 167)) w.r.t. n

(n - 168, fact (n - 168)) w.r.t. n

(n - 169, fact (n - 169)) w.r.t. n

(n - 170, fact (n - 170)) w.r.t. n

(n - 171, fact (n - 171)) w.r.t. n

(n - 172, fact (n - 172)) w.r.t. n

(n - 173, fact (n - 173)) w.r.t. n

(n - 174, fact (n - 174)) w.r.t. n

(n - 175, fact (n - 175)) w.r.t. n

(n - 176, fact (n - 176)) w.r.t. n

(n - 177, fact (n - 177)) w.r.t. n

(n - 178, fact (n - 178)) w.r.t. n

(n - 179, fact (n - 179)) w.r.t. n

(n - 180, fact (n - 180)) w.r.t. n

(n - 181, fact (n - 181)) w.r.t. n

(n - 182, fact (n - 182)) w.r.t. n

(n - 183, fact (n - 183)) w.r.t. n

(n - 184, fact (n - 184)) w.r.t. n

(n - 185, fact (n - 185)) w.r.t. n

(n - 186, fact (n - 186)) w.r.t. n

(n - 187, fact (n - 187)) w.r.t. n

(n - 188, fact (n - 188)) w.r.t. n

(n - 189, fact (n - 189)) w.r.t. n

(n - 190, fact (n - 190)) w.r.t. n

(n - 191, fact (n - 191)) w.r.t. n

(n - 192, fact (n - 192)) w.r.t. n

(n - 193, fact (n - 193)) w.r.t. n

problem

power ( $n, y$ ) /  $n^y$

$$n^y = \begin{cases} 1; & y=0 \\ n \cdot n^{(y-1)}; & \end{cases}$$

$$\begin{aligned} 3^5 &= 3 \times 3^4 \\ &= 3 \times 3 \times 3^3 \\ &\approx 3 \times 3 \times 3 - 3 \times 3 \\ &= 3 \times 3 \times 3^0 \rightarrow 1 \end{aligned}$$

#include<stdio.h>

int pow(int, int); // (n^y, "b^n") function

int main()

```
{
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d ^ %d = %d", a, b, pow(a, b));
}
```

int pow(int, n, int y); // (n^y, "b^n") function

```
{
    if (y == 0) return 1;
    return n * pow(n, y-1);
}
```

pow (5, 3)

5 \* pow (5, 2)

5 \* pow (5, 1)

5 \* pow (5, 0)

## problem

print the sum all the natural number upto N

```
#include <stdio.h>           <N.oibz> ab05n#  
int sum (int);             : (hai) is correct too!  
  
int main ()                (miss tri  
{.                         {  
    int n;                   (o tri )  
    scanf ("%d", &n);        ((0, "6") too?  
    printf ("Sum of first %d = %d", n, sum(n));  
}  
  
int sum (int n)            {  
{  
    if (n == 1) return 1;    (o == n) tri  
    return n + sum(n-1);    (1 == n) tri  
    return n + sum(n-1);    ((n-1) is correct + (1-n) is modif correct  
}; ((n-1) is correct + (1-n) is modif correct
```

$$S = 1 + 0 + 1$$

|   |   |   |   |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
|---|---|---|---|

(c) is correct  
 $(1 \text{ bit} + 1 \text{ bit})$   
 $1010101010101010$

## ① Fibonacci

```
#include <stdio.h>  
int fibonacci (int);
```

```
int main()
```

```
{  
    int a;
```

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

```
((r)) printf ("%d", fibonacci(a)); fibo
```

```
}
```

```
int fibonacci (int n)
```

```
{
```

```
if (n==0) return 0;
```

```
if (n==1) return 1;
```

```
return fibonacci(n-1) + fibonacci(n-2);
```

```
}
```

fibonacci (3)

$$\underline{\text{fib}(2)} + \underline{\text{fib}(1)}$$

$$\underline{\text{fib}(1)} + \underline{\text{fib}(0)}$$

$$1+0+1 = 2$$

|   |   |   |   |
|---|---|---|---|
| 0 | 1 | 1 | 2 |
| 0 | 1 | 1 | 2 |

problem

If ( $y == 0$ ) return  $n$ ;  
 $\text{GCD}(n, y) \rightarrow \text{GCD}(y, n \% y)$

find GCD using recursion

#include <stdio.h>

int GCD(int); int);

int main()

{

int a, b;

scanf("%d %d", &a, &b);

printf("GCD = %d", GCD(a, b));

}

int GCD(int n, int y)

{

if ( $y == 0$ ) return  $n$ ;

return GCD(y, n % y);

}

$$\begin{array}{r} 3) \quad \begin{array}{c} 17 \\ 15 \end{array} \\ \overline{y \quad 2) \quad \begin{array}{c} 3 \\ 1 \end{array}} \\ \begin{array}{c} 1 \\ 2 \\ 2 \\ \hline 0 \end{array} \end{array}$$

GCD

tugri

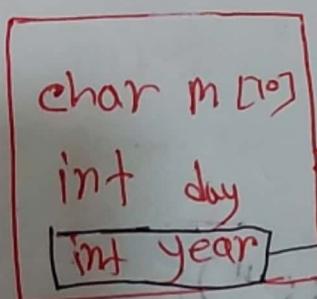
## structure

R.N.Bible & S.Balaji #

Array के लिए सबसे Data type आएगा at a time

C supports data structures that can store combination of character, integer, float and double.

These are called a struct.



structure

member variable

minimum 2 रास्ते Data type आएंगे गये

(root, rot3)

("stab base rabbit")

rot3, pub, rot3, "bx bx br")

(rot3, b3, chrom, h3)

input

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

struct Date

```
{  
    int day;  
    int month;  
    int year;  
};
```

```
int main()
```

struct Date

```
    start, end;
```

printf ("Enter start date: ");

```
scanf ("%d %d %d", &start.day, &start.month  
    &start.year);
```

```
printf ("Enter end date: ");
```

```
scanf ("%d %d %d", &end.day, &end.month,  
    &end.year);
```

```
printf("Start Date : %d-%d-%d\n", start.day,  
      start.month, start.year);
```

```
printf("End date : %d-%d-%d\n", end.day,  
      end.month, end.year);
```

```
return 0;
```

```
}
```

input

20 10 2021  
20 9 2023

(Date) A

(Date) B

<National Holiday #  
Output

20-10-2021  
20-9-2023



typedef struct

{

int day;  
int month;  
int year;

}

Date;

Data type

id 3 Friday

another way

declare to  
struct.

problem

write down a program that defines a structure to store n and Y coordinate of a point in 2-dimensional space. then take n,y coordinate of two points as user input

A  $\overrightarrow{(n, y_1)}$  B  $(n_2, y_2)$

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

```
typedef struct
```

```
{  
    float n;  
    float y;  
} point;
```

```
int main()
```

```
{  
    point a, b;
```

```
    scanf("%f %f", &a.n, &a.y);
```

```
    scanf("%f %f", &b.n, &b.y);
```

float distance;

$$\text{distance} = \sqrt{\text{pow}(a.x - b.x, 2) + \text{pow}(a.y - b.y, 2)}$$

printf("%f", distance);

return 0;

}

\* Create a structure named structure that has name, roll, mark, and grade as member.

Find his/her grade using a user defined

function

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

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

```
typedef struct
```

```
{
```

```
char name[30];
```

```
int id;
```

```
float mark;
```

```
} student;
```

Output student (st)

char check\_grade (student);  
int main ()

{

student st; // input = student

printf ("Enter name: ");

gets (st.name);

printf ("Enter id: ");

scanf ("%d", &st.id);

printf ("Enter mark: ");

scanf ("%f", &st.mark);

st.grade = check\_grade (st);

printf ("name : %s\n", st.name);

printf ("Id: %d\n", st.id);

printf ("mark: %f\n", st.mark);

printf ("grad : %c\n", st.grade);

counts kept }

char check-grade (students)

{

m = mark;

char g;

if ( s.m >= 90 ) ~~<= 100~~ g = 'A';

else if ( s.m >= 80 & s.m <= 90 ) g = 'B';

else if ( s.m >= 70 & s.m <= 80 ) g = 'C';

else if ( s.m >= 60 & s.m <= 70 ) g = 'D';

else g = 'F';

} return g;

## multiple student GR अंतर्गत

typedef struct

{

char name[30];

int id;

float mark;

char grade;

} student;

char check\_grade (student);

int main ()

{

student std [n]

for (int i=0; i<n; i++)

{

gets (std[i].name);

scanf ("%d", &std[i].id);

scanf ("%f", &std[i].mark);

scanf ("%c", &std[i].grade);

std[i].grade = check\_grade (std[i]);

}

```

    printf("Name: %.5s\n", std[i].name);
    for (int i=0; i<n; i++)
    {
        printf ("Name: %.5s\n", std[i+1].name);
        printf ("Id: %.5d\n", std[i+1].id);
        printf ("Mark: %.2f\n", std[i].marks);
        printf ("Grade: %.2c\n", std[i].grade);
    }
}

char check_grade (student a)
{
    char g;
    if (a.marks>=90) g='A';
    else if (a.marks>=80) g='B';
    else if (a.marks>=70) g='C';
    else if (a.marks>=60) g='D';
    else g='F';
    return g;
}

```

## Q. Pointer

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

```
int main()
```

```
{ int a;
```

```
a = 10;
```

```
printf ("%d %d\n", a, &a);
```

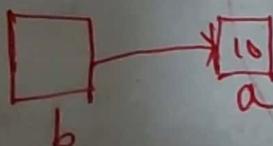
```
return 0;
```

value      address

pointer گھر marker تکمیل کیا type نہ

```
int a = 10, *b;
```

```
b = &a;
```

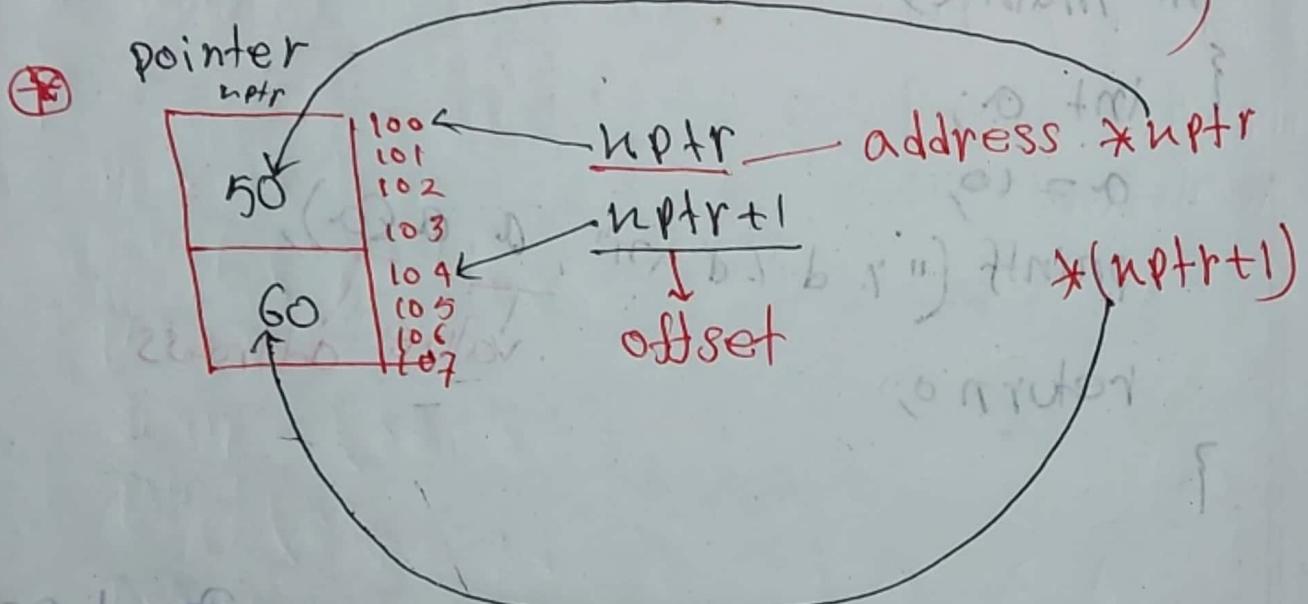


```
printf ("%d %d\n", b, a, *b); output 10, 10
```

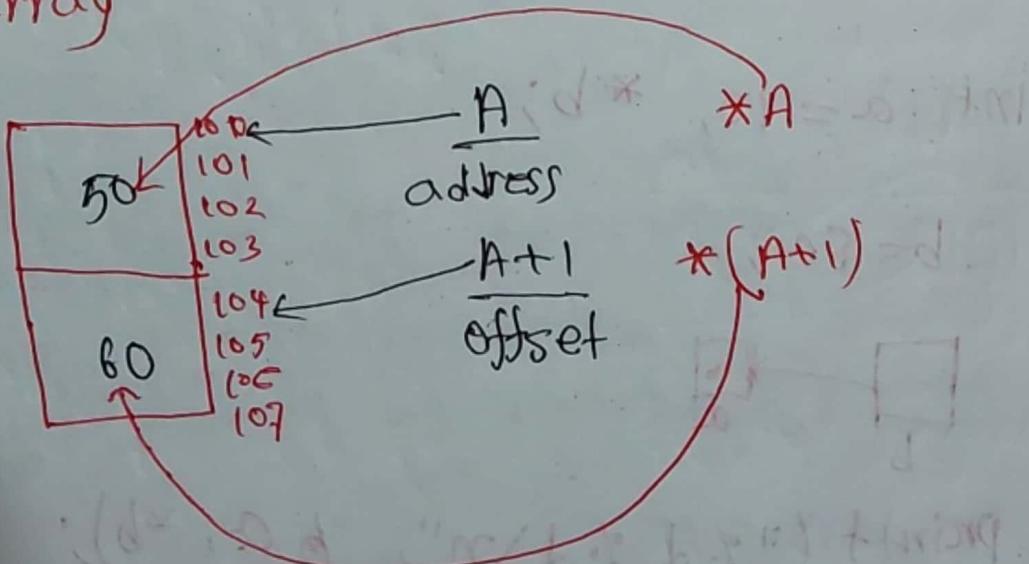
```
scanf ("%d %d", &a, b); *b = 50
```

`int *p;  
*p = 100` } error

`int *p, a = 10;` } right  
 $*p = 80$   
 $*p = 100$



⊕ array



$$*(b+1) = 20$$

$$*b + 1 = 11$$

| b  | 0 | $b[0] = *b$     |
|----|---|-----------------|
| 10 | 1 | $b[1] = *(b+1)$ |
| 20 | 2 | $b[2] = *(b+2)$ |
| 30 | 3 | $b[3] = *(b+3)$ |
| 40 |   |                 |

White down a program that will take two integer as input using pointer then using pointer multiply the two values and display the result.

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

```
int main ()
```

```
{
```

```
    int a, b;
```

```
    int *p, *q;
```

```
    p = &a;
```

```
    q = &b;
```

```
    scanf ("%d", p); // scanf ("%d", &a)
```

```
    scanf ("%d", q); // scanf ("%d", &b);
```

```
    int result = (*p) * (*q); // result = a * b;
```

```
}
```

pointer function यादें void कर्ता लगाए

swap

void

#include <stdio.h>

void swap(int, int);

int main()

{

int a, b;

scanf("%d%d", &a, &b);

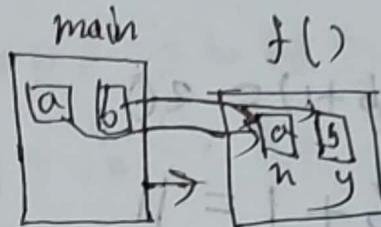
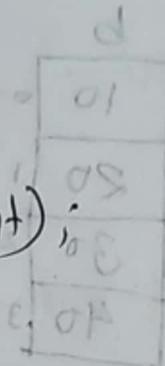
printf

swap = (a, b);

printf("a: %d, b: %d\n", a, b);

return 0;

}



copy किए यादू रखो

function for main

swap किए यादू रखो

void swap(int \*x, int \*y)

{

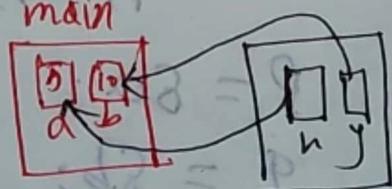
int t;

t = \*x;

\*x = \*y;

\*y = t;

}



pointer original value

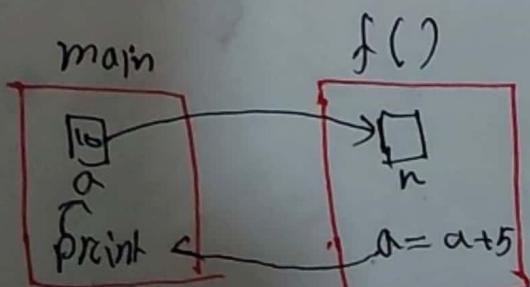
किए यादू रखो

```

#include <stdio.h>
void print( int &n)
{
    n = n + 5;
}
int main()
{
    a = 10;
    printf(" %d ", a);
    print(a);
    printf(" %d ", a);
}

```

Output 10, 10



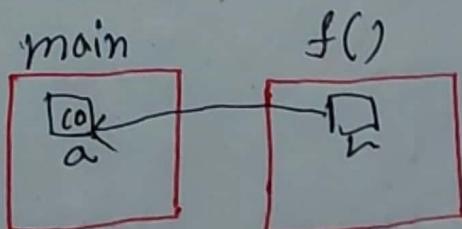
COPY করা অঙ্গ হল main  
Get উন্নিষ্ঠিত value change করুন

```

#include <stdio.h>
Void print( int *n)
{
    *n =
    *n = *n + 5;
}
int main()
{
    a=10;
    printf(" %d ", a);
    print(&a);
    printf(" %d ", a);
    return 0;
}

```

Output: 10, 15



original value কে কোথা  
করুন, return করুন address  
value কি, কোথা change  
করুন। এটা কোথা

## File / Text file

- ① open the file, with file name
- ② Read or write the data.
- ③ close the file.

```
#include <stdio.h>
int main ()
{
    FILE *fp
    fp = fopen ("data.txt", "w");
    if (fp == NULL)
    {
        printf ("Error");
        return;
    }
    printf ("Successfully opened ");
}
```

file create  $\Rightarrow$

file write  $\Rightarrow$

file read  $\Rightarrow$

~~white mode~~

file পাঠ্য এবং Run দেখলে ফলট কী  
কোর্ট কী না।

#include <stdio.h>

int main ()

{

FILE \* p

p = fopen ("Rakib.txt", "w");

if (p == NULL) fine name

whi  
file কোর্ট

{

printf ("Error. ");

return ;

}

printf ("successfully opened \n")

{ float a = 10;

float b = 3.14;

char c = 'A';

fprintf (p, "%d %f %c \n", a, b, c);

|       |
|-------|
| Rakib |
| 10    |
| 3.14  |
| A     |

return 0;

}

important

Rakib

P = fopen ("Rakib.txt", "w");

10. 3. 19 A

নতুন প্রক্রিয়া করা হবে

P = fopen ("Rakib.txt", "a");

10. 3. 19 A  
10. 3. 19 A

বাস্তু স্থানে মন্তব্য

নতুন - প্রক্রিয়া করা (পুরো  
পুরো)

⊗⊗⊗ White mood G শর্করা file অন্তর্ভুক্ত  
শর্করা file এবং শর্করা { নাম (Rakib.txt)}  
By default file নাম রকিব।

⊗⊗⊗ Read mood G শর্করা অন্তর্ভুক্ত শর্করা file  
না নামে খুলে Error দেয়।

Read mode

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

```
int main()
```

```
{ FILE *T;
```

```
T = fopen("Rating.txt", "r");
```

```
if (T == NULL)
```

```
{
```

```
    printf("Error");
```

```
    return 0;
```

```
}
```

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

```
int a;
```

```
float b;
```

```
char c;
```

```
fscanf(T, "%d %f %c", &a, &b, &c);
```

```
printf("%d %f %c", a, b, c);
```

```
return 0;
```

```
}
```

file

10, 3.19 A

Rating.txt

display

10, 3.19 A

Rating.txt

(10, 3.19 A)

जाना file का अंदर का क्या है

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

```
int main()
```

```
{
```

```
FILE (*R)
```

```
R = fopen ("student.txt", "r");
```

```
if (R == NULL)
```

```
{
```

```
printf ("Error");
```

```
return;
```

```
}
```

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

```
char name[30];
```

while(1)

```
{
```

```
if (feof (R))
```

```
{
```

```
break;
```

```
}
```

```
fscanf (R, "%s", &name);
```

```
puts (name);
```

```
}
```

File R

Rakib  
Tamim

Display

Rakib  
tamim

Rahib Hussain  
0112310308

Name: Rahib Hussain

Student ID: 0112310308

## ONLINE I (SET C)

Write a C program that will take as input an integer, replace its last digit with its first one, and then find out if the new number is a palindrome or not.

| Sample input | Sample output             |
|--------------|---------------------------|
| 89471        | 89478<br>Not a palindrome |
| 5178715      | 5178715<br>Palindrome ✓   |

89471

1) 2 1 3 4  
7 4 9 8

$8 \times 10^4$

= 80000

8 7 4 9 8

Ans: [100];

