

## structure

```
#include <stdio.h>
int main ()
{
    // all your code here
    printf (" - - ");
    return 0;
}
```

\* \n → নতুন লাইন যাওয়া যাবে।

\* \t → কোন কম্পিউটারের 8/64 টি space.

\* // → এটা Comment Compiler এর কোন কাজ করবে না।

## DATA TYPE and variables

চলক যাই তার পরিমাণালী।

Unsigned → কোন negative Limit থাকবে না।

\* Variable name soft word এর রক্ষণ রেজ।

## Memory size

integer → 4 type byte.

double type data → 8 type byte

character → 1 byte

Placeholder → অন্ত হাতে ফায়েজ করেন।

%d ⇒ integer.

↳ কোর্স value রে ফর্ম্যাট করে এবং।

Character  
↳

%c ⇒ a value কেন্দ্রে প্রেরণ করে।

%f = fraction number.

integer and long integer

\* #include <stdio.h>

```
int main () {  
    int a;  
    int b;  
    a=10;  
    b=3;  
    printf ("Result in : %d\n", a+b);  
    return 0;  
}
```

↳ modulus / remainder / অবশেষ।

\* long long integer → 8 byte → 64 bit  
↳ integer so. limit extra 112 → overflow  
↳ long long integer → integer so. limit extra 112 → overflow + check 32bit  
↳ long long integer → 8 byte → 64 bit  
↳ long long integer → 8 byte → 64 bit  
↳ long long integer → 8 byte → 64 bit  
↳ long long integer → 8 byte → 64 bit

To find out the limit of my competitor

```
#include <stdio.h>
int main()
{
    printf("size of int in my computer: %d\n", sizeof(int));
    return 0;
}
```

To find the limit of  $\frac{1}{n}$  as  $n$  increases.

~~# print ("The maximum value of INT = %d\n", INT\_MAX);~~

Printf ("The maximum value of INT = %d \n", INT\_MAX);  
printf ("The minimum value of Long = %ld \n", LONG\_MIN);

11 GHOST THEM EXAMPLES

`#include <stdio.h>` → It header file provides various functions for input and output.

卷之三

radio  $\rightarrow$  standard input and output  
radio detail not often problematic - assumes radio

July 26, 1965.

$\text{G}_{\text{H}_2\text{S}} = \text{G}_{\text{H}_2} + \text{G}_{\text{S}}$

float / - - अकाले ना मिलता ।

## Float and Double: float वह डॉट का बहुत #

float → single precision floating point number → 4 byte  
double → double precision floating point number. → 8 byte  
from float to double printing  
printf ("Value of a: %.f\n", a); → float  
printf ("product: %.lf\n" . . . ); → lf → double

## Character and ASCII Value: float वह डॉट का बहुत # क्षणीय तथा नियम संरक्षण करने के लिए इसका उपयोग किया जाता है।

char my\_char;

my\_char = 'K';

my\_char = '3';

my\_char = 'U';

my\_char = 'M';

my\_char = 'A';

my\_char = '1';

my\_char = '0';

my\_char = '2';

my\_char = '5';

my\_char = '7';

my\_char = '9';

my\_char = '8';

my\_char = '6';

my\_char = '4';

my\_char = '3';

my\_char = '2';

my\_char = '1';

→ It will show my\_char = '3'  
→ It will show my\_char = 'M'  
→ It will show my\_char = 'K';  
→ It will show my\_char = 'U';  
→ It will show my\_char = 'N';  
→ It will show my\_char = 'A';  
→ It will show my\_char = '1';  
→ It will show my\_char = '0';  
→ It will show my\_char = '2';  
→ It will show my\_char = '5';  
→ It will show my\_char = '7';  
→ It will show my\_char = '9';  
→ It will show my\_char = '8';  
→ It will show my\_char = '6';  
→ It will show my\_char = '4';  
→ It will show my\_char = '3';  
→ It will show my\_char = '2';  
→ It will show my\_char = '1';

ASCII → अंग्रेजी वर्षाकारी अकाले ना मिलता ।  
ASCII code / value:  
→ इन्हें key press करने से गेट होता है।  
ASCII stands for American Standard Code for Information

Interchange:

Char - Code (int).

space - 32.      a - 97.  
A - 65

\* char x = 'A';

```
printf("%c\n", x);
```

→ (mathematical)

~~char~~  $X = 'A'$ ;

printf("%d(%c,%c)", integer number X goes here, ASCII number or print code 1, 1964)

\* int a = 65, b = 66, c = 77;  
printf("%c %c %c\n", a, b, c);

## User Input Using scanf()

```

char d;
scanf ("%c", &d);
int X; float Y;
scanf ("%d %f", &X, &Y);
float my_float;
scanf ("%f", &my_float);
double my_double;
scanf ("%lf", &my_double);

```

# float x, y;

printf ("Enter value for x and y : ");  
scanf ("%f,%f", &x, &y);  
printf ("The value of x is %f and y is %f.",  
x, y);

Condition: If  $= \Sigma$  the

Condition: If - Else

if ( condition ) {  
// If the condition is true - - .  
// then these statements will work . . .  
} // End of if block

```
int x = 5;  
if (x == 5) {  
    printf ("value of x is %d.\n"),
```

```

} } // after final result.
int x = 200;
if (x > 100) {
    printf ("%d\n", x);
}

```

$$x = 25$$

```

if (x > 100) {
    printf ("Hi (%d);\n");
    printf ("Ting (%d);\n");
}

```

January 20, 1945 - 1000 hours

Chair শ্রদ্ধালুর single coat 27821 ।

$\rightarrow$   $\text{outgoing}$

Char mychar = 'X';

$$\{R_i = \text{achieved}\}$$

Printed ( $\text{Hi} \backslash \text{m}$ ):

```
print f("ting \n");
```

卷之三

If (Condition) {

11 If the condition in true - - .

// then these statements will work --

3

三  
九

III If the condition in tab

// then these statements will work -- -

It follows that  $\alpha = \beta$  and  $\gamma = \delta$ .

bullock (1905) describes 12 genera of the tribe (Fig. 3, 0.52).

\* int x = 5;

卷之三

شیوه ایجاد

26

print ("Hello \n");

print ("He \n");

2

$\neq \rightarrow$  not equal.

Foot

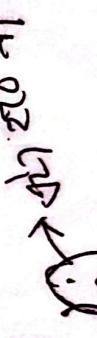
\* \* One Conditional statement will work at a time.

Q1:

```

char ch = 'a';
if (ch == 'a') {
    printf ("Apple");
}
else if (ch == 'b') {
    printf ("Banana");
}
else if (ch == 'c') {
    printf ("Cherry");
}
else {
    printf ("Unknown");
}

```

~~the if (- - -)~~  if  $x = x + A'$   $\{x\} \text{ at last}$   $i = 1$   $S_{\text{end}}$  using "main" procedure

1. "A Day in the Life" (1980) 2. "The Big Picture" (1981) 3. "The Big Picture" (1982)

• In concave value function.  $\rightarrow$   $f''(x) < 0$   $\rightarrow$   $f'(x)$  is decreasing.

break fast প্রস্তুত করা হয়।

switch (n) {

Care Care - i:

```
// Code to execute if n == case - 1
```

Case Case - 2:

11 case to execute if  $n = \text{case} - 2$   
break;

卷之三

```
// Code to execute if l == case - i (l = 1 to n) It will  
break;  
default:  
// default :- if no case matches
```

10

卷之三

$\sin(\theta)(x) \}$

卷之三

- 2 -

unit (Value 1) → break,

case 2:  
Printf ("Value in : %n");

case 3:  
print f (" value in : 3 \n");

switch keyword: switch, case, break, default.

```
default:  
printf ("Value in: unknown \n");  
char ch = 'C'; if (ch  
switch (ch) {  
case 'a':  
    printf ("Value in: a\n");  
    break;  
case 'b':  
    printf ("Value in: b\n");  
    break;  
case 'c':  
    printf ("Value in: c\n");  
    break;  
default:  
    printf ("Value in: unknown \n");  
}
```

Explanation:

- default: This keyword is used to define the code block that will be executed if none of the cases in the switch statement matches the value of the variable being compared.
- char ch = 'C'; if (ch: This part of the code initializes a character variable ch and assigns it the value 'C'. It then checks if the value of ch is true (which it is, since 'C' is not equal to 0).
- switch (ch) { This starts the switch statement, where the value of ch is used as the condition for the cases.
- case 'a': This is the first case in the switch statement. It checks if ch is equal to 'a'. Since ch is 'C', this case does not match.
- printf ("Value in: a\n");: If the case matched, this line would have been printed.
- break: This keyword is used to exit the current case and move to the next case or the default case.
- case 'b': This is the second case in the switch statement. It checks if ch is equal to 'b'. Since ch is 'C', this case does not match.
- printf ("Value in: b\n");: If the case matched, this line would have been printed.
- break: This keyword is used to exit the current case and move to the next case or the default case.
- case 'c': This is the third case in the switch statement. It checks if ch is equal to 'c'. Since ch is 'C', this case does not match.
- printf ("Value in: c\n");: If the case matched, this line would have been printed.
- break: This keyword is used to exit the current case and move to the next case or the default case.
- default:  
 printf ("Value in: unknown \n");: This is the default case, which is executed because none of the previous cases matched. It prints the string "Value in: unknown \n".

Signature & Syntax:

```
(case, default, switch, break)
```

switch statement syntax:

```
switch (expression) {  
    case constant1:  
        statements  
    case constant2:  
        statements  
    ...  
    default:  
        statements  
}
```

~~Example-1~~

```
int a = 1;
if (a == 1) {
    printf("Hi\n");
} else if (a == 2) {
    printf("Hello\n");
} else {
    printf("Good Bye\n");
}
default:
    printf("Grad Bye\n");
}
System.out.println("Java");
```

Operators, Precedence and Associativity:

$x = y + z$  - operator  $\Rightarrow +$ ,  $=$ ; operand :  $x, y, z$ .

Arithmetical operator : These are the operators used to perform arithmetic/mathematical operations on operands.

Arithmetical operator are of two types: 1. Unary 2. Binary

1. Unary 2. Binary  
→ operators that operate on works with a single operand are unary operators. ( $z++$ ,  $-x$ )  
→ operators that operates on works with two operands are binary operators. For example ( $t, -t, \sqrt{t}$ )

Relational: are used for comparison of the values of two

operands. ( $=, \geq, >, <, \leq$ ).

Logical: are used to combine two or more conditions/ constraints or to complement the evaluation of the original condition in consideration.

Assignment: are used to assign value to a variable.

( $=, +, -, *, /, \%, \&$ )

( $* \Rightarrow + \rightarrow - \rightarrow \div \rightarrow \%$ )  $\Rightarrow$  precedence of operators like  $*$  and  $/$  is same.

int a = 5;

a += 10;  $\rightarrow$  a = a + 10 (addition,  $+=$  assignment)

--a  $\rightarrow$  1 becomes zero  
a -= 1  $\rightarrow$  1 becomes zero

++a  $\rightarrow$  1 becomes one  
a ++  $\rightarrow$  2 becomes one

(5 > 10 - 5 885)  $\rightarrow$  False.

$!= \rightarrow$  not equal.

(1 == 2 != 0)  $\rightarrow$  False.

```

graph TD
    loop[loop] --> whileLoopDo[while loop do]
    forLoopDo[for loop do] --> loop
    loop --- grouping1[looping]
    whileLoopDo --- grouping1
    forLoopDo --- grouping1
    grouping1 --- text1[looping is used for repeating a block of statements]
    whileLoopDo --- grouping2[loop]
    forLoopDo --- grouping2
    grouping2 --- text2[looping is used for repeating a block of statements]
  
```

```

for (initialization; condition; increment/decrement) {
    // Code to execute
}

int i;
for (i = 1; i <= 5; i++) {
    printf("%d\n");
}

```

```

int a;
for (a=1; a<=5; a=a+1) {
    printf("%d\n", a);
}

```

length tail = 1.

```
#include <iostream>
using namespace std;
int i;
for (i = 1; i <= 3; i++) {
    cout << i;
}
cout << endl;
```

1.  $i = 1$  starting  
2.  $i = 2$  starting  
3.  $i = 3$  starting  
out of loop  
cout ("out of loop\n");

loop এর  
ক্ষেত্রে যা  
প্রক্রস.

```
#include <iostream>
using namespace std;
int i;
for (i = 3; i >= 1; i--) {
    cout << i;
}
cout << endl;
```

1.  $i = 3$  starting  
2.  $i = 2$  starting  
3.  $i = 1$  starting  
out of loop

\* Mane a summation of between 2 to 525 which are  
divided by 2, 5 or 10.

while Loop

```
initialization
while (Condition) {
    // code to execute
    increment /decrement
}
```

cout << i;

$i = 1$

$i = 2$

$i = 3$

$i = 4$

$i = 5$

$i = 6$

$i = 7$

$i = 8$

$i = 9$

$i = 10$

$i = 11$

$i = 12$

$i = 13$

$i = 14$

$i = 15$

$i = 16$

$i = 17$

$i = 18$

$i = 19$

```
#include <iostream>
using namespace std;
int main()
{
    int x = 1;
    while (x <= 3)
    {
        cout << "Hi I'm ";
        cout << x << endl;
        cout << "Hello I'm ";
        x = x + 2;
    }
}
```

Do while loop → यहाँ जारी दो तरीके हैं।  
1. logic false रखें और अवधि लिया जाए।  
2. logic true पर भी अवधि लिया जाए।

### initialization

```
do
{
    // code to execute
    increment / decrement
}
while (condition);
```

प्रकारिति (count) + minimum

### advantages

1. (condition) की ओर नहीं जाना होता।  
2. कोड की संरक्षण करना आसान होता।

### disadvantages

```
int i;
i = 1;
do
{
    cout << "Let me go! " ;
    i = i + 1;
}
while (i <= 2);
```

## Nested loop → फॉर लॉप में फॉर लॉप का इसी तरह होता

```
int i, j;
for(i=1; i<=3; i++) {
    printf("Thin in outer loop");
    for(j=1; j<=2; j++) {
        printf(" Thin in inner loop");
    }
}

** Int i, j;
for(i=1; i<=2; i++) {
    printf("outer loop starts\n");
    for(j=1; j<=3; j++) {
        printf(" * * * inner loop\n");
    }
    printf("outer loop ends\n");
}

** int i, j, k;
for(i=1; i<=2; i++) {
    for(j=1; j<=3; j++) {
        for(k=1; k<=5; k++) {
            printf(" * \n");
        }
    }
}
```

→ Good job no work

print thin paper

$\{ \alpha_i \}_{i=1}^n$  使得  $\sum_i \alpha_i = 1$  且  $\alpha_i \geq 0$  对所有  $i$  成立。

→ outer loop for line no. 0 to  $M-1$ ; inner loop for  $i=1$  to  $M$ .

( $j = 5; j \geq i; j--$ )  
( $j = 1; j < s; j++$ )  
for sum of first  $j$  terms + last

## Break and Continue

```
int i;
for (i=0; i<3; i++) {
    printf("Hi\n");
    if (i == 1)
        continue;
    printf("Hello\n");
}
```

output of  
loop

Now we can write  
 $\log_{10}(1 + x) = \log_{10}(1 + 10^{-n})$   
 $= \log_{10}(1 + 10^{-n} + 10^{-n} + \dots)$   
 $= \log_{10}(1 + 10^{-n}) + \log_{10}(1 + 10^{-n}) + \dots$   
 $= n \log_{10}(1 + 10^{-n})$

while(1)  $\Rightarrow$  शब्द वर्त न पढ़ सके।

## Array Introduction in C, S1 = {J N H}

एक ऐरेला एवं एक फिलर लिंग मानिया जाएगा तथा एवं आवश्यक

Array इसका लो।

int x;  $\rightarrow$  float variable  $\rightarrow$  1 वर्ग डेटा ग्रूप के लिए float

int x [3];  $\rightarrow$  float array  $\rightarrow$  3 वर्ग डेटा ग्रूप float

x[3] = x[0], x[1], x[2]

Array / डायरेक्ट्रीमिंडरेक्ट्री

\* int a[3],

a[0] = 3,

a[1] = 4;

a[2] = 400;

int x = a[0] + a[2];

$\Rightarrow$  403

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

a[2] = 100;

x = a[0] + a[2]

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

scanf → अंतिम प्रिंटर का info. (1) 10/10

\* int a[5] = {12, 11, 13, 20, 32};

printf ("Value of index 0: %d\n", a[0]);

\* int a[5] = {13, 11, 13, 20, 32};

int i;

for (i=0; i<5; i++) {

printf ("%d\n", a[i]);

}

\* int a[5];

int i;

for (i=0; i<5; i++) {

scanf ("%d", &a[i]);

for (i=0; i<5; i++) {

printf ("%d\n", a[i]);

## Array Operation

\* int a[7] = {-99, 45, 100, 37, 89, -327, 245};

int max = a[0];

for(i=1; i<7; i++) {

if(a[i] > max)

max = a[i];

}

\* int a[5], b[5], ab[5];

int i;

printf("Enter the value of A: \n");

for(i=0; i<5; i++) {

scanf("%d", &a[i]);

printf("Enter the value of B: \n");

for(i=0; i<5; i++) {

scanf("%d", &b[i]);

}

{ (max = b[0]) b[0] to max (i) ) + max

; (max = max, max = b[1], min max to max) + max

```

for(i=0; i<5 ; i++) {
    ab[i] = a[i] + b[i];
}
printf ("Value of AB array: \n");
for(i=0; i<5; i++) {
    printf("%d", ab[i]);
}
int a[5], i, even-sum = 0, odd-sum = 0;
for(i=0; i<5; i++) {
    scanf("%d", &a[i]);
}
printf ("Values of Array: ");
for(i=0 ; i<5; i++) {
    printf("%d", a[i]);
}
if(a[i] % 2 == 0)
    even-sum = even-sum + a[i];
else
    odd-sum = odd-sum + a[i];
printf ("sum of odd values: %d \n", odd-sum);
printf ("sum of even values: %d \n", even-sum);

```

~~exception~~

## Two Dimensional (2D) Array in C

DATA TYPE Array Name [row] [column];

```
int x[2][3] = {11, 12, 13, 21, 22, 23};
```

```
printf("%d", x[1][2]); // output: 23
```

11	12	13
21	22	23

```
int a[3][3], i, j;
```

```
for (i=0; i<3; i++) {
```

```
    for (j=0; j<3; j++) {
```

```
        printf("%d", a[i][j]);
```

```
    }
```

```
    for (i=0; i<3; i++) {
```

```
        for (j=0; j<3; j++) {
```

```
            printf("%d", a[i][j]);
```

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

~~for~~

1495943

## Function

```
int get_sum (int x, int y) { } [इस फंक्शन का नाम जॉर्ज एवं एम्पी है]
```

int sum = x + y;  
return sum;  
}

## Conditional operator / ternary operator

num1 ? num2 : num1.

```
{  
    int num1, num2, large;  
    printf ("Enter two numbers:");  
    scanf ("%d%d", &num1, &num2);  
    large = (num1 > num2) ? num1 : num2;
```

प्रथम वाला  
गलत जानकारी  
हो गया

large = (num1 > num2) ? num1 : num2;

जल्दी बदला

जल्दी बदला

जल्दी बदला

जल्दी बदला

जल्दी बदला

large = (num1 > num2) ? num1 : num2;

large = (num1 > num2) ? num1 : num2;

गलत

## Bitwise Operators

(उपरोक्त की प्रयोग में)

→ Can only be used on integers, don't work with float.

→ name input & output 0

& → Bitwise AND / | → Bitwise OR / | → Bitwise EXOR / >> → Right shift  
<< → Left shift / ~ → Bitwise NOT.      → name input & output 0

## Loop

for, while, do while

```
{ int i;  
    for( i=1 ; i<=5 ; i++ )  
    { printf( "C Programming \n" );  
    }  
    return 0; }
```

```
{ int i;  
    for ( i=1 ; i <= 100 ; i+4 )  
    { printf( "%d \n" , i );  
        printf( "say +a" );  
    }  
    return 0; }
```

### while loop

```

    {
        int i=1;
        do cout << i << endl;
        while (i<=10)
        {
            cout << endl;
            cout << "print (%d\n", i);
            cout << endl;
            i++;
        }
    }

```

### Do while

```

    {
        int i=1;
        do
        {
            cout << i << endl;
            cout << endl;
            cout << "print (%d\n", i);
            cout << endl;
            i++;
        } while (i<=10);
        return 0;
    }

```

## Loop

### while

```

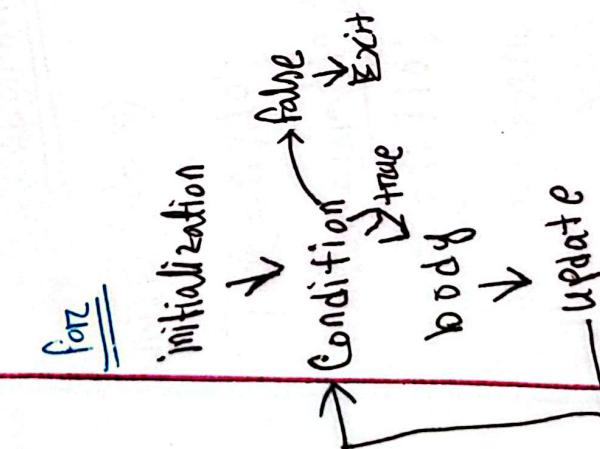
{int i;
for(i=1; i<=100; i++)
{printf("%d\n", i);
}
    
```

### do while

```

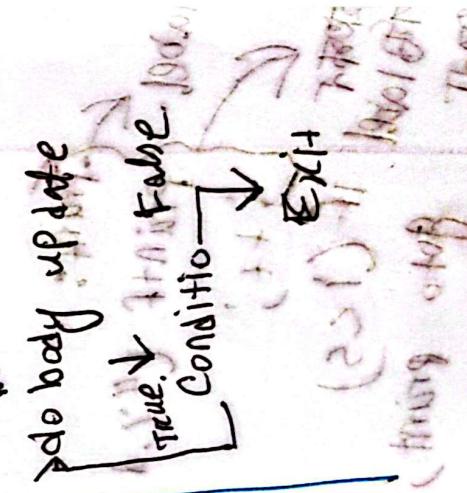
{int i;
i=1;
do {i++;
    printf("%d\n", i);
    i++;
}
while (i<=100);
    
```

### Flow chart



### while

### do while



## Continue/Break

```

int i = 1;
for (i = 1; i < 20; i++)
{
    if (i % 3 == 0)
        continue;
    printf("%d\n", i);
    if (i == 10)
        break;
}

```

```

int i;
for (i = 1; i <= 10; i++)
{
    if (i % 3 == 0)
        continue;
    printf("%d\n", i);
}

```

```

int i;
for (i = 1; i <= 10; i++)
{
    if (i % 3 == 0)
        continue;
    printf("%d\n", i);
}

```

Note:

```

int i = 1;
for (i = 1; i <= 10; i++)
{
    printf("%d\n", i);
    if (i < 5)
        goto print;
    i++;
}
print:
    label:
    printf("i = %d\n", i);
    goto label;
    print:
    printf("i = %d\n", i);
    i++;
    if (i < 5)
        goto print;
    i++;
}

```

অসম অসম অসম



ଅନ୍ତରେ କାଳେ ମଧ୍ୟରେ ରେବେରେ ହାତ୍ୟାକାଣ୍ଡ ପାଦ୍ସୁନ୍ଦର ଶାରୀ ଯିମେ ଯାଏ  
ତଥାଲେ କେବଳ ପାଠିନ୍ଦ୍ରମେ ନାହିଁ ।

ତଥାଲେ କୌଣସିବାରେ Palindrome number.  
# Armstrong number → ଏହା ସଂଖ୍ୟା କରାଯାଇଲେ ଯାଥିରେ ପରିମାତ୍ର ପରିମାତ୍ର ହେବାରେ ଏହା କାହାରେ ଦିଆଯାଇଲେ ତାହାରେ ଜାହେଲ୍ଲାରେ କିମ୍ବା କିମ୍ବା କିମ୍ବା

$1234 \rightarrow 1^4 + 2^4 + 3^4 + 4^4 = 1234$  是 armstrong  
 $\text{while } (n! = 0) \rightarrow \text{while } (n! \neq 0)$

```

    n = 1/5;
    ++ count;
}
}

```

$$\text{sum} = \text{r}^{\wedge} \text{Count} + \text{sum};$$

$$\text{sum} = \pi \wedge \text{Count} + \delta$$

$$10 \quad \begin{array}{r} 12 \\ \times 23 \\ \hline 293 \end{array}$$

3

Step 3

$$n = 2 \frac{1}{2}(a) \quad RL = 1 \quad 5.16m^3 + 3^3 + 1^3.$$

$$\begin{array}{l} \text{Sum} = 3^3 + 0, \\ \text{Sum} = 27, \\ \text{Sum} = 2+3, \\ \text{Sum} = 5, \\ n=1 \end{array}$$

12

2

1/5

400 X 500 = 2000

2000-01

11 11  
W W  
A A  
O O  
E E

7

51

$$e + \alpha x_3 = m_2$$

10  
co  
ey

卷之三

1/5

A 28

卷之三

10

400 X 500 = 2000

111

7

## \* Counting number of a digit.

```
while(n!=0)
```

```
{ n = n/10;  
  ++count; }
```

\* Strong number ÷ strong digit क्षमता factorial एकान्तरण ओर  
number तक strong number।

$$145 = 1 + 4 + 5$$

$$= 145$$

```
int num, rem = 0, fact = 1;  
char c;  
fact = 1;
```

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

```
{ fact = fact * i; }
```

```
if(fact == num)  
{ cout << "Strong Number"; }
```

```
else if(fact > num)  
{ cout << "Over"; }
```

```
else if(fact < num)  
{ cout << "Under"; }
```

```
else  
{ cout << "Normal"; }
```

$$1 \cdot 2 + 2 \cdot 4 + 2 \cdot 3 = 12,$$

$$2 + 4 + 6 = 12$$

Row 001

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

1 2 3 4 5

2 \*

1 ..

3 ..

4 ..

5 ..

initial O = min, 0 = max, \* 2 + 3, \*

4 = min, 0 = (min, max) 2 + 3, \*

(0 = 1 min, 1) 3 + 4, \*

0.1 min = 0.95, \*

1.1 max = 1.05, \*

1.1 \* max = 1.05, \*

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	1	2	3	4	5
4	4	1	2	3	5
5	5	4	3	2	1

	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5






</tbl\_struct

Ques. কোডটা যেনের প্রক্রিয়া কী?

```
int temp;
temp = x;
x = y;
y = temp;

for(i = 10; i < 20; i++)
{
    for(j = 2; j <= i; j++)
    {
        if(i == j)
            break;
        if(i == j)
            printf("%d", j);
    }
}
```

প্রক্রিয়া হলো

- 1.  $i = 10$  হলে  $j = 2$  থেকে  $i$  পর্যন্ত একটি লুপ চালা হবে।
- 2.  $i = 11$  হলে  $j = 2, 3$  থেকে  $i$  পর্যন্ত একটি লুপ চালা হবে।
- 3.  $i = 12$  হলে  $j = 2, 3, 4$  থেকে  $i$  পর্যন্ত একটি লুপ চালা হবে।
- 4.  $i = 13$  হলে  $j = 2, 3, 4, 5$  থেকে  $i$  পর্যন্ত একটি লুপ চালা হবে।
- 5.  $i = 14$  হলে  $j = 2, 3, 4, 5, 6$  থেকে  $i$  পর্যন্ত একটি লুপ চালা হবে।
- 6.  $i = 15$  হলে  $j = 2, 3, 4, 5, 6, 7$  থেকে  $i$  পর্যন্ত একটি লুপ চালা হবে।
- 7.  $i = 16$  হলে  $j = 2, 3, 4, 5, 6, 7, 8$  থেকে  $i$  পর্যন্ত একটি লুপ চালা হবে।
- 8.  $i = 17$  হলে  $j = 2, 3, 4, 5, 6, 7, 8, 9$  থেকে  $i$  পর্যন্ত একটি লুপ চালা হবে।
- 9.  $i = 18$  হলে  $j = 2, 3, 4, 5, 6, 7, 8, 9, 10$  থেকে  $i$  পর্যন্ত একটি লুপ চালা হবে।
- 10.  $i = 19$  হলে  $j = 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$  থেকে  $i$  পর্যন্ত একটি লুপ চালা হবে।
- 11.  $i = 20$  হলে  $j = 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12$  থেকে  $i$  পর্যন্ত একটি লুপ চালা হবে।

String → character type array.

scanf → space is no part of string. program; one gets user into 200/gets (5); string name is stored.

```
print("I am a string");
```

\* Chain [T] = " শাখা " → যে মে কণ রং line branch  
" শাখা " ;

```
#include <stdio.h>
int main ()
{
    char s1 [30];
    printf (" Enter your full name: ");
    gets (s1);
    printf (" Full name = %s , s1 );
    return 0;
}
```

string value print int, char

int main()

{ char t [] = "Udit";

int i = 0;

while { t[i] != '\0' } → start at index 0 to value of character

```
{ printf ("%s\n", t);  
i++; }
```

}

string length function finds exact string length upto first function strlent, read  
char t [] = "Udit" Sarita Sarita;  
int j = 0; len = 0;  
while (t[j] != '\0') → count till it reaches '\0'  
int len = strlen(t);  
printf ("length = %d\n", len);  
len++;

{ }

Copy source to target copy করে সেকেন্ড স্ট্রিং তৈরি

( ) নিচের ফার্ম

Function এর সামগ্র্য,

char source[] = "গুলু";

char target[20];

strcpy(target, source);

printf("target string = %s", target);

char target[] = " ";

(0 = i + n)

target[i] = source[i];

{ (i) নথি হলে target[i] = ' ' } ফল

একটি স্ট্রিং এর সাথে সারেন করে স্ট্রিং (ভিত্তি) এর

Function এর সামগ্র্য,

char str1[] = "গুলু";

char str2[] = "গুলু";

strcat(str1, str2);

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

(i) নথি হলে str1[i] = ' '

(i) নথি হলে str1[i] = ' '

char str1[] = "গুলু";

strcat(str1, "গুলু");

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

target[i] = source[i];

(i) নথি হলে target[i] = ' ' } ফল

(i) নথি হলে target[i] = ' ' } ফল

Function used are CTRN and I.

char s1[] = "महाराष्ट्र";

(A. write) Shabab #

char s2[] = "गोपनी";

(B. Read + HI)

int i = 0, len = 0, j = 0;

while(s1[i] != '\0')

{ i++;

len++;

}

while(s2[j] != '\0')

(B. Read)

int k = 0;

{ s1[len + j] = s2[k];

(A. write) Shabab #

j++; }

(B. Read + HI)

printf("strc = %.5s", s1);

(A. Read + HI)

(B. Read + HI)

(A. Read + HI)

(B. Read + HI)

"गोपनी गोपनी गोपनी गोपनी" का अर्थ

(max. 6N) Max

(MIN) Excpn = the str - HI

{ (Max. (N/6N) \* group) } का अर्थ

## Function

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

Return type      function-name (arguments)

```
{    return a+b;
```

Int      sum

(int a, int b)

प्रतिक्रिया  
जागति

```
#include<stdio.h>
int square (int a);
{ return a*a;
}
```

int main()

```
{ int num;
printf("Enter any integer number:");
scanf("%d", &num);
int result = square(num);
printf("square is : %d\n", result); }
```

1.  $\text{sum}(1,2) \rightarrow \text{sum}(1,2) = 1+2$

2.  $\text{sum}(1,2) \rightarrow \text{sum}(1,2) = 1+2$

$a=5, b=10, c=15$

$\text{sum}(1,2) = [1,2] = 1+2$

$\{ \quad \} = 3$

#include <stdio.h>  
double triangleArea(double b, double h);

int main()

```
{ double base, height;
scanf("%lf", &base);
scanf("%lf", &height);
```

double area = triangleArea(base, height);

printf("Area = %.lf\n", area);

}

double triangleArea(double b, double h)

```
{ return 0.5 * b * h; }
```

Recursion: এ ক্ষেত্রে ফাংশন নিয়ে বিজ্ঞেতা করলে এটি একটি পুরো ফাংশন।

Recursive call এর সময় ক্ষেত্রে নিয়ে বিজ্ঞেতা করলে এটি একটি ফাংশন।

Base case এর সময় করলে এটি একটি condition এর জন্ম।

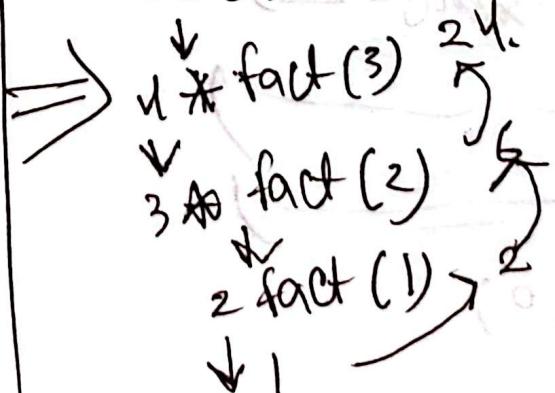
fact(4)

fact(int n)

```
{ if (n == 1)
```

```
    return 1;
```

```
else
    fact(n-1);}
```



\* main function  $\rightarrow$  access global structure  $\rightarrow$  global variable  
\* main n II Proto. Local II

structure  $\rightarrow$  X-139

30am C

\* collection of different data type under a single name  
\* A structure is a user defined data type in C/C++

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

```
struct person
```

```
{ int age;
```

```
 float salary;
```

```
} ;
```

```
int main()
```

```
{ struct person person1, person2;
```

```
person1.age = [ ];
```

```
person1.salary = [ ];
```

```
printf(
```

```
person2.age = [ ];
```

```
person2.salary = [ ];
```

```
printf(
```

```
return 0;
```

local variable function proto access from main

global variable

## Structure Compare:

No point in ~~copying~~ structure variable Pass  
partial program from one function to another.

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

```
struct person {  
    char name[20];  
    int age;  
    float salary;};
```

```
void display(struct person p)
```

```
{ printf("Name: %s \n", p.name);  
printf("Age: %d \n", p.age);}
```

```
int main()
```

```
{ struct person person1;  
person1.name = "Animal Islam";  
person1.age = 27;
```

```
person1.salary = 12250.50;
```

```
display(person1);
```

```
return 0;
```

```
}
```

## Union

Union is a user defined data type. In union, all members share the same memory location.

struct test

```
{ char ch;
  int x;
  float y;
}
```

memory size  
= 1 + 4 + 4 = 9

Union test.

```
{ char ch;
```

int x;

float y;

{  
 9 memory type (char) for char  
 4 memory type (int) for int  
 4 memory type (float) for float  
 = 1 + 4 + 4 = 9 bytes

If we want to use same memory for two more members.

#include <stdio.h>

Union test {

int x, y;

}

int main()

{ Union test t;

t.y = 10;

printf("%d", t.x);

printf("%d", t.y); }

$$\left. \begin{array}{l} t.y = 10 \\ t.x = 10 \end{array} \right\}$$

$y.u$  = decimal value.

$y.x$  = hexadecimal value.

Enum: An enumeration is a user-defined data type that consists of integral constants.

Pointer →  $\&$  type  $\&int$  Point  $\&int$  Point is  $\&$  type data  $\&int$ .

Pointer is a variable that stores / points the address of another variable.

int  $\&p$  → pointer variable

$\&$  symbol

$\&$  symbol is used to get the address of the variable.

$*$  symbol is used to get the value of the variable that the pointer is pointing to.

$\&\&$

Variable name →  $x$        $y$

Value → 5      10

Variable address → 100      104

int  $x = 5;$

int  $y = 10;$

int  $\&p;$

$p = \&x;$

printf ("Value of  $x = \%d$  \n",  $x$ );

printf ("Address of  $x = \%d$  \n",  $\&x$ );

printf ("Address of  $P = \%d$  \n",  $p$ );

printf ("Content of  $P = \%d$  \n", \* $p$ );

\* memory address always positive  $\&$  variable  
- i.e. we  $\&$  to  $\&v$ . U use  $\&v$  to  $v$ .  
printf ("Address of  $P$  = \%d",  $p$ );  
= i.e.  $\&100$ ,  $\&5$ ;

निम्नों तरीके से बांधा जाता है। **FILE** एवं उसके प्रयोग का विवर।

## **FILE**

निम्नों हमें दिये गए हैं।

• File is a place on disk where a group of related data is stored.

• File is a structure which is stored in file header.

Declaration of File:

FILE \* name-of-file → file pointer variable

To write something in a file | To read something in a file.

fputc()

fgetc()

fputw()

fgetw()

fputs()

fgetcs()

fprintf()

fscanf()

fwrite()

fread()

#include <stdio.h>

int main()

{ FILE \*file;

fopen("f", "r");

file =

file so mode.

fclose(file);

close file.

## File mode

r → file already exists & read only

w → file from user created file

a → new file creation

(X) X = 9

b X = X to sub

(X) X = X to memba

(X) X = 9 to memba

(X) X = 9 to memba

# subscript index variable.

↳ A data structure in a particular way of storing and organizing data in a computer.

# concatenation = the process of combining two strings together

# atoi converts a string digit to integer value.

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

```
int main ()
```

```
{ int result = fact(5);
```

```
printf ("Result"; result);
```

```
}
```

```
int fact (int n)
```

```
{ if (n == 1)
```

```
return 1;
```

```
else,
```

```
return = n * fact(n-1);
```

```
}
```

String 5, 6  
Matrix multiplying so all operations about A &  
Matrix multiply  
for (i=0; i < n1; i++)  
{ for(j=0; j < C2; j++)  
    { for(k=0; k < C1; k++)

$$\left\{ \text{sum} = \text{sum} + \text{A}[i][j] * \text{B}[j][k] \right.$$

( ) n1 m + n1

for i = 1 to n1  
for j = 1 to n2

if (i == 1)  
    return i;

else  
    return i + (i - 1)

(i == n1) {

    i = n1 + 1  
    return i;

else  
    return i + n1 + 1

else  
    return i + n1 + 1

\* `int fibo(int z, int y)`

```
static int i = 1;  
int term,  
    x(i == n);  
return 0;
```

+ 6.50

```

int main()
{
    int i, fact;
    float x;
    cout << "Enter a number: ";
    cin >> x;
    cout << "Factorial of " << x << " is ";
    fact = fact(x);
    cout << fact;
}

float fact(float n)
{
    if (n == 0)
        return 1;
    else
        return n * fact(n - 1);
}

```

```

int fact (int n) {
    if (n == 1) return 1;
    else return n * fact(n-1);
}

```