DATA TYPES

- Different types of data to be stored in memory. Eg. integer, float, character, double, etc

Eg - int → Stores integers like -5, 0, 8 etc.
char → single character like 'a', '+', '$', '7', etc.
float → Floating point value like -2.014, 1.000, 6.78 etc.

- Different data types use different amount of memory. Amount of memory used also depends on the architecture of your CPU.

Datatypes	Size (in Bytes)
int	2 or 4
float	4
double	8
char	1
wchar_t (wide character)	2
bool	1
void	0

```
character :   char ch = 'a';
Boolean   :   bool isGood = 1;
              bool isBad = false;

Float     :   float num1 = 1.2;
Double    :   double num2 = 2.4;
```

Variable Naming / Nomenclature:

1. Can contain alphabets, numbers and underscore
2. Cannot start with a number.
3. Cannot be keywords like int, count, double, bool, etc.
4. Case sensitive
5. Cannot contain special symbols like %, $, !, #, etc.

=> To check the size of different Data Types for your system using.

$\boxed{\text{sizeof(variable\_name);}}$

=> what if we store a negative value in an unsigned number?

Example

```
unsigned int a = -112;
cout << a << endl;
```

output => 4294967184

Explanation

We tried to store -112

-112 = 2's complement of 112

$112 \Rightarrow \underbrace{00\ldots0}_{25\text{ zero}}1110000$

$$
\begin{array}{rr}
1\text{'s compl.} = & 11\ldots10001111 \\
+ & 1 \\
\hline
2\text{'s comple} = & 11\ldots10010000 \\
\hline
\end{array}
$$

- Unsigned int uses all 32 bits to store the value and the MSB (=1) will make the value.
- As unsigned int does not use the 2's complement again to display the number.

Thus, 11...10010000 gets printed as it is in decimal

Basic Arithmetic Operators :

+, - , * , / , %

① int / int = int (Floor' value of answer)

Example = 5/2 = 1
3/5 = 0
9/2 = 4

② int / float } float
float / int }

cout << 5.0/2 << endl
outputs 2.5

double / int } double
int / double }

Relational Operators :

== , >= , <= , >, <, !=

① Is a=b ?

$a == b \xrightarrow{Yes} 1$

$\quad \xrightarrow{No} 0$

② Is a greater than or equal to b ?

$a >= b \xrightarrow{Yes} 1$

$\quad \xrightarrow{No} 0$

Logical Operators

&& , || , !
(AND) (OR) (NOT)

While Loop

```
while (condition) → while the condition is true,
{                   keep on executing the block.

}
```

Example : Print 1 to N

```
int main()
{    int n
     cin>>n;
     int i=1;
     while (i <=n) {
          cout <<i<< " ";
          i = i+1;
     }
     return 0;
}
```

output : 6
1, 2 3 4 5 6

Example : Find sum from 1 to N

```
int main()
{  int n;
  cin>>n;
  int sum =0;
  int i = 1;
  while (i <=n){
  sum = sum+i;
    i = i+1;
  cout << "Sum from" << 1
  <<"to "<<n<<"=" <<sum;
  return 0
}
```

output : 8
sum from 1 to 8 =36

Example: Sum of all even no. from 1 to N

```
int main()
{
   int n;
   cin>>n;
   int sum=0;
   int i=2;
   while (i<=n) {
       if (i%2==0)
            sum = sum + i;
         i = i + 1;
   }
   cout<<"Sum from"<<1<<"to"<<n<<"="<<sum<< endl;
   return 0;
}
```

output 8
sum from 1 to 8 is =20

Example : Prime or not

```
int main()
{    int n;
     cin>>n;
     int i= 2;
     while(i<n) {
        if(n%i==0) {
           cout<< "Not Prime" << endl;
           return0;
        }
        i = i+1
     }
    cout << "Prime" << endl;
    return 0;
}
```

output 14
Not prime