```
//! Data Types (sizes/range)
    /*
    ! note : the number of bytes defernet depending on
            ~ 1) the operating system  32 bit or 64 bit
            ~ 2) type of operating system
    % note: by default everything is signed
        ? int: (integer)
            # the size in Byte : 4
            range -2147483648 to 2147483647
            ^ it get 2 value negative and positive
        ? Float: (floating point)
            # the size in Byte : 4
            range 1.17549e-38 to 3.40282e+38
            ^ it get 2 value negative and positive
        ? doble: (double floating point)
            # the size in Byte : 8
            range 2.22507e-308 to 1.79769e+308
            ^ it get 2 value negative and positive
        ? char: (character)
            # the size in Byte : 1
            range -127 to 127
        ? wchar_t:  (wide character)
            # the size in Byte : 2
        ? bool: (boolean)
            # the size in Byte : 1
            range 0 (false) or 1 (true)
        ? void: (empty)
            # the size in Byte : 0
        ? string:
            # the size in Byte : 12

    ! Type Modifiers:
        ? it is used to modify some of fundimental data types (int , double , char)
        # there are 4 type modifiers in c++
            ~ 1) signed
                $ int
                    * by default signed that mean yiu can use positive and negative number
                    * the range of signed int -2147483648 to 2147483647
                    * the size of signed int 4 bytes
                    ^ note : the + / - character store inside the last bit in 4 byte
```

```
~ 2) unsigned
    % use it when you want to get more space/ range without using additional byte
    $ unsigned int
        # it used to store positive nuber just
        * the get last byte inside the 4 byte use it for number
        ! so the range will be bigger from (0 to 4294967295)
~ 3) short
    $ short int / short
        ^ it is signed short int by default
        ^ the size of signed short int 2 bytes
        ^ the range from -32768 to 32767
    $ unsigned short int / short
        ^ the size of unsigned short int 2 bytes
        * the get last byte inside the 4 byte use it for number
        ^ so the range will be bigger from  0 to 65536
~ 4) long
    $ long int
        ^ it is signed long int  by default
        ^ the size of signed long int 4 bytes
        ^ the range from -2147483648 to 2147483647
    $ unsigned long int
        ^ the size of unsigned long int 4 bytes
        * the get last byte inside the 4 byte use it for number
        ^ so the range will be bigger from  0 to 4294967295
~ 5) long long
    $ long long int
        ^ it is signed long long int  by default
        ^ the size of signed long int 8 bytes
        ^ the range from -(2^63) to  (2^63)-1
    $ unsigned long long int
        ^ the size of unsigned long long int 8 bytes
        * the get last byte inside the 8 byte use it for number
        ^ so the range will be bigger from  0 to 18446744073709551651
~ 6) float
    $ its signed and cannot be unsigned
    ^ the size of it 4 byte
    % the range 1175449e-38 to 340282e+38
~ 7) double
    $ its signed and cannot be unsigned
    ^ the size of it 8 byte
    % the range 222507e-308 to 179769e+308
~ 8) long double
    $ its signed and cannot be unsigned
    ^ the size of it 12 byte
    % the range 10 e-307 to 10e+308
```

```
                ~ 9) char
                    $ signed char
                        ? its by default
                        ? the size 1 byte
                        ? the range from -127 to 127
                    $ unsigned char
                        ? the size 1 byte
                        ? the range from 0 to 255
                ~ 10) bool
                    ^ the size of it 1 byte
                ~ 11) string
                    ^ the size of it 12 byte
                ~ 12) void (empty)
                        ^ the size of it 0 byte

            % notes:
             ?   #1E#2 -->    #1 * 10 ^ #2
             ?   if you writ number number out of range the data type it will cause an error
             #   (overflow) mean that error out of range
             ~   in some compiler if you write wrong value like:
                    $ out of range
                    $ or assign negative value inside signed variable
                        * it will give rubbsh data

          ! to get the size of data type: use sizeof(dataType) method
             ? sizeof(bool) //1
             ? sizeof(char)  // 1
             ? sizeof(short int)  //2
             ? sizeof(int)  // 4
             ? sizeof(long)  //4
             ? sizeof(long long)  // 8
             ? sizeof(float) // 4
             ? sizeof(double) // 8

        ! to konw the range of the data type use the dataType_MIN and dataType_MAX
            $ cout<< "char Range :"<< CHAR_MIN <<"," << CHAR_MAX ;
            $ cout<< "unsigned char Range :"<< 0 <<"," << UCHAR_MAX ;


            $  cout<< "short int Range :"<< SHRT_MIN <<"," << SHRT_MAX ;
            $ cout<< "unsigned short int Range :"<< 0 <<"," << USHRT_MAX ;

            $ cout<< "int Range :"<< INT_MIN <<"," << INT_MAX ;
            $ cout<< "unsigned int Range :"<< 0 <<"," << UINT_MAX ;
            $ cout<< "long int Range :"<< LONG_MIN <<"," << LONG_MAX ;
            $ cout<< "unsigned long int Range :"<< 0 <<"," << ULONG_MAX ;

            $ cout<< "long long int Range :"<< LLONG_MIN <<"," << LLONG_MAX ;
            $ cout<< "unsigned long long int Range :"<< 0 <<"," << ULLONG_MAX ;


            $ cout<< "float Range :"<< FLT_MIN <<"," << FLT_MAX ;
           $ cout<< "float (negative) Range :"<< -FLT_MIN <<"," << -FLT_MAX ;

            $ cout<< "double Range :"<< DBL_MIN <<"," << DBL_MAX ;
            $ cout<< "double (negative) Range :"<< -DBL_MIN <<"," << -DBL_MAX ;
            $ cout<< "long double Range :"<< LDBL_MIN_10_EXP <<"," << LDBL_MAX_10_EXP ;
```