



Data Types And Operators

C ක්‍රමලේඛනයේදී ,දත්ත වර්ගය යනු යම් විචල්‍යක, දත්ත වර්ගය හා පරමාණය තීරණය කිරීමට යොදාගනු ලබන පරකාශනය වේ .

int price;

මෙහි price යනු int වර්ගයට අයත් විචල්‍යකි.මෙම විචල්‍යයේ ගබඩා කළ හැකි දත්ත වල විශාලත්වය බයිට(Byte) 4 කි.

Main Data Types Use In C Language.

Type	Size (bytes)	Format Specifier
int	at least 2, usually 4	%d
char	1	%c
float	4	%f
double	8	%lf
short int	2 usually	%hd
unsigned int	at least 2, usually 4	%u
long int	at least 4, usually 8	%li
long long int	at least 8	%lli
unsigned long int	at least 4	%lu
unsigned long long int	at least 8	%llu
signed char	1	%c
unsigned char	1	%c
long double	at least 10, usually 12 or 16	%Lf

int Data Type

int යනු ශුන්ය, ධන හා සෘණ වන පූර්ණ සංඛ්‍යා ගබඩා කිරීමට යොදා ගන්නා දත්ත වර්ගය වේ.

int price;

C භාෂාවේදී එකවර බහු විචල්‍යයන් ප්රකාශ කිරීමටද හැකියාව ඇත .මෙය ක්රමලේඛනය ඉක්මන් කිරීමට බෙහෙවින් උපකාර වේ.උදාහරණයක් ලෙස,

int price,total;

int දත්ත වර්ගයේ ප්රමාණය බයිට් 4 (බිට් 32) ක් වේ.මේ නිසා -2147483648 සිට 2147483647 දක්වා අගය පරාසයන් තුළ ඕනෑම සංඛ්‍යාවක් විචල්‍යකට පැවරිය හැකි වේ.

float And double data types.

මෙම දත්ත වර්ග දෙකම සංඛ්‍යා ගබඩා කිරීමට යොදා ගනු ලැබේ.

float salary;

double number;

What is difference of float and double types:

float වලදී බයිට් 4 ක දත්ත ගබඩා කිරීමට හැකි

විමන්,double වලදී මෙම දත්ත ප්රමාණය දෙගුණයක්

නැතහොත් බයිට් 8 ක් ගබඩා කළ හැකිවීමත් මෙම දත්ත

වර්ග දෙකෙහි ප්රධානතම වෙනස වේ.

Char Data type.

මෙම දත්ත වර්ගය අකුරු ආකාරයේ දත්ත රඳවා තබා ගැනීමට යොදා ගනී.

මෙම දත්ත වර්ගයෙන් ගබඩා කළ හැකි දත්ත වල විශාලත්වය බයිට් 1 කි.

Ex:

char letter;

char deshan;

char lion;

short හා long දත්ත වර්ග.

අපට විශාල සංඛ්‍යාවක් ගබඩා කිරීමට අවශ්‍ය වූ විටෙක මෙම **long** දත්ත වර්ගය යොදාගත හැකි වේ.

එසේම කුඩා අගයක් ගබඩා කිරීමට අවශ්‍ය නම් **short** යන දත්ත වර්ගය යොදා ගැනේ.

long a;

long long b;

long double c;

ඔයාලට මේ එක් එක් දත්ත වර්ග වල ගබඩා කළ හැකි උපරිම දත්ත ප්රමාණය දැන ගන්න ඔනි කියල හිතන්න , එහෙනම් එකට ලේසියෙන්ම **sizeof()** යන operator එක භාවිතා කරන්න පුළුවන්.

```
prog.c X
1  #include <stdio.h>
2  int main(){
3      int a;
4      long b;
5      long long c;
6      short d;
7      char e;
8      double f;
9      float g;
10     long double h;
11
12     printf("size of int type = %d bytes\n",sizeof(a));
13
14     printf("size of long type= %d bytes\n",sizeof(b));
15
16     printf("size of long long type= %d bytes\n",sizeof(c));
17
18     printf("size of short type= %d bytes\n",sizeof(d));
19
20     printf("size of char type= %d bytes\n",sizeof(e));
21
22     printf("size of double type= %d bytes\n",sizeof(f));
23
24     printf("size of float type= %d bytes\n",sizeof(g));
25
26     printf("size of long double type= %d bytes\n",sizeof(h));
27
28 }
29
```

මේ code එක run කරලා බලන්න එතකොට පහලින් පෙන්නලා තියෙන විදියේ output එකක් බලාගන්න පුළුවන් වෙයි.(prog.c)

```
"C:\Users\Jayashanka Deshan\Desktop\prog.exe"
size of int type = 4 bytes
size of long type= 4 bytes
size of long long type= 8 bytes
size of short type= 2 bytes
size of char type= 1 bytes
size of double type= 8 bytes
size of float type= 4 bytes
size of long double type= 12 bytes

Process returned 0 (0x0)   execution time : 0.089 s
Press any key to continue.
```

C Operators.

operator එකක් යනු විචල්යයක් මත හෝ අගයක් මත යම් ක්‍රියාකාරකමක් කිරීමට භාවිත කරන සංකේතයක් වේ. උදාහරණයක් ලෙස (+) සංකේතය මගින් අගයන් කීපයක් එකතු කිරීමට භාවිත කරයි. තවත් උදාහරණයක් දෙන්නම් ,

a හා b කියල variable දෙකකට 10 හා 20 කියල අගයන් දෙකක් පවරලා තියෙන අවස්ථාවක් සලකමු, මේ අගයන් දෙකෙන් විශාලතම අගය සොයාගන්න (>) යන සංකේතයද කුඩා අගය සොයාගන්න (<) සංකේතයද යොදා ගන්න පුළුවන් .

C භාෂාවේදී , ඉතා විශාල පරාසයක විභිද්‍යන operators ප්රමාණයක් භාවිතා වෙනවා. අපි මේවට උදාහරණ කීපයක් සලකා බලමු.

- A. Arithmetic operators.(ගණිතමය ක්‍රියාකාරකයන්)
- B. Increment හා Decrement operator.
- C. Assignment operators.(පැවැරුම් ක්‍රියාකාරකයන්)
- D. Relational operators.(සම්භන්දනා ක්‍රියාකාරකයන්)
- E. Logical operators.(තාර්කික ක්‍රියාකාරකයන්)

A.Arithmetic operators.(ගණිතමය ක්‍රියාකාරකයන්)

ගණිතමය ක්‍රියාකාරකයන් යනු , එකතුවීම , අඩුකිරීම , බෙදීම , වැඩිකිරීම වැනි අංක ගණිතමය ක්‍රියාකාරකම් සිදුකිරීමට භාවිත කරන operators වේ.

Operator	Meaning of Operator
+	addition or unary plus
-	subtraction or unary minus
*	multiplication
/	division
%	remainder after division (modulo division)

```
#include<stdio.h>
int main() {
    int a=50,b=30,anw;
    //addition
    anw=a+b;
    printf(" answer is = %d\n",anw);

    //subtraction
    anw=a-b;
    printf("answer is = %d\n",anw);

    //division
    anw=a/b;
    printf("answer is = %d\n",anw);

    //multiplication
    anw=a*b;
    printf("answer is = %d\n",anw);

    return 0;
}
```

CODE	OUTPUT
<pre> int main(){ int a=50,b=30,anw; //addition anw=a+b; printf(" answer is = %d\n",anw); //subtraction anw=a-b; printf("answer is = %d\n",anw); //division anw=a/b; printf("answer is = %d\n",anw); //multiplication anw=a*b; printf("answer is = %d\n",anw); return 0; } </pre>	<pre> "C:\Users\Jayashanka Deshan\Desktop\prog2.exe" answer is = 80 answer is = 20 answer is = 1 answer is = 1500 Process returned 0 (0x0) execution time : 0.058 s Press any key to continue. </pre>

B.Increment And Decrement operator.

increment (++)	decrement (–)
යම් විචල්‍යක අගය එකකින් වැඩි කිරීමට	යම් විචල්‍යක අගය එකකින් අඩු කිරීමට

Ex:prog3.c

code	output
------	--------

```
#include<stdio.h>
int main(){
    int num1=10,num2=100;
    float
    num3=10.5,num4=100.5;

    printf("++num1 =
%d\n",++num1);

    printf("--num2 = %d\n",-
num2);

    printf("++num3 =
%f\n",++num3);

    printf("++num4 = %f\n",-
num4);

    return 0;
}
```

```
"C:\Users\Jayashanka Deshan\Desktop\prog3.exe"
++num1 = 11
--num2 = 10
++num3 = 11.500000
++num4 = 99.500000

Process returned 0 (0x0)   execution time : 0.020 s
Press any key to continue.
```

Edited by Jayashankaha deshan