Data type:

There is a difference between integer and data type Int.

Int x; is this integer?

X can not store these values 29272856726784234;

Hence x is not of type integer.

We know it has a range of 32 bits in which 1 is a sign bit and the other 31 are data bits.

Data type

int x; -> x int

int x[10] -> x int[10]

int x[10][5]; -> x int[10][5]

char m -> m char

char m[10] -> m char[10]

char m[10][20] -> m char[10][20]

sizeof - operator to determine the size of the data type. At the compile time.

#include<stdio.h>

int main(){

int x;

int y[5];

int z[10][20];

char m;

char k[5];

char g[10][20];

printf("%d\n", sizeof(x)); //4

printf("%d\n", sizeof(y)); //20

printf("%d\n", sizeof(z)); //800

printf("%d\n", sizeof(m)); //1

printf("%d\n", sizeof(k)); //5

printf("%d\n", sizeof(g)); //200

printf("%d\n", sizeof(int)); //4

return 0;

}

This above code if we run in windows no issue

But in mac or ubuntu we will get warning: but we can compile code.

-> format specifies type 'int' but the argument has type 'unsigned long' [-Wformat]

printf("%d\n", sizeof(int));

We can use lu in place of d, lu means long unsigned int

Mac  
#include<stdio.h>

int main(){

int x;

int y[5];

int z[10][20];

char m;

char k[5];

char g[10][20];

printf("%lu\n", sizeof(long unsigned int)); //8

printf("%lu\n", sizeof(x)); //4

printf("%lu\n", sizeof(y)); //20

printf("%lu\n", sizeof(z)); //800

printf("%lu\n", sizeof(m)); //1

printf("%lu\n", sizeof(k)); //5

printf("%lu\n", sizeof(g)); //200

printf("%lu\n", sizeof(int)); //4

return 0;

}

**Sizeof is not a function its operator that can be used to determine the size of any given data type.**

**It will work at the time of compilation not at the time of execution.**

But when see print sizeof address

#include<stdio.h>

int main(){

int x;

int y[5];

int z[10][20];

char m;

char k[5];

char g[10][20];

// printf("%d\n", sizeof(x)); //8

printf("%lu\n", sizeof(&x)); //8

printf("%lu\n", sizeof(&y)); //8

printf("%lu\n", sizeof(&z)); //8

printf("%lu\n", sizeof(&m)); //8

printf("%lu\n", sizeof(&k)); //8

printf("%lu\n", sizeof(&g)); //8

return 0;

}