Initializing Array

Int x[3];

Int x[3] = { 10,20,30}; // Correct

Int x[] = { 10,20,30}; // Correct

Int x[5] = { 10,20,30}; // Correct BUT

// As per c23 Standard Gcc V11 and above will approve this and make remaining 0

But we should consider it as garbage(is suppose any oid system, we have to use can not not reply on that)

Int x[10] = {}; // all elements will be zero

Int x[10] = { [ 0 … 9 ] = 2030 }; // SPACE BEFORE AND AFTER … IS REQUIRED.

INT X[10] = { [2] = 10, 20, 30 };

0

0

10

20

30

0

0

0…

It means from 2nd inder it will start placing the values.

INT X[10] = { 0, [2 … 4 ] = 10, 20, 30, 0 };

0

0

10

10

10

20

30

0

0

0

INT X[5] = { [3 ] = 10, []0 … 2 ]= 50 };

50

50

50

10

0

Example of 2D array

Int x[4][3] = {

{ },

{ },

{ },

{ }

}

Initialise in one line.

#include<stdio.h>

int main(){

int x[4][3] = { 10,20,30 , 40,50,60, 70,80,90,100,110,120 };

int r,c;

for(r=0; r<4 ; r++){

for(c=0; c<3 ; c++){

printf("%4d", x[r][c]);

}

printf("\n");

}

return 0;

}

/\*

10 20 30

40 50 60

70 80 90

100 110 120

We can intialize in one line but to print we need two variables r & c

\*/

Case of Dynamic 2d

We can not assign a[][];

#include<stdio.h>

int main(){

// the follwing Declation is incorrect

// first can be empty but renst can not be

// becase we can put dufferent number of values in dirrect row

int x[][] = {

{10,20,30},

{40,50,60},

{70,80,90,100}

};

int r,c;

for(r=0; r<4 ; r++){

for(c=0; c<3 ; c++){

printf("%4d", x[r][c]);

}

printf("\n");

}

return 0;

}

/\*

error: array has incomplete element type 'int[]'

int x[][] = {

^

1 error generated.

\*/

We can only pass 1st element blank rest we have to put

X[][] ;// not allowed

X[][3] ; // alliowed/s

For more

Go to Google

Search

Empty initialization in c

en.cppreference.com

Or c23 standard

If we initialzise global array then all elements will be zero

#include<stdio.h>

int x[4];

int main(){

int r;

for(r=0; r<3 ; r++){

printf("%4d\n", x[r]);

}

return 0;

}

/\*

0

0

0

0

\*/