How to get a sequence of regular indices in C?

(such as the sequence [2 6 10], starting at 2 and the step is 4 and go through 3 times)

[intro]

Using the class -- slice.

slice is a special class that achieve this.

In the slice class,

if you want to get the start value, call its start() method.

If you want to get the step, call the stride() method.

If you want to get the number of elements, call the size() method.

|  |  |
| --- | --- |
| Method | Note |
| start() | get the start value |
| stride() | get the step |
| size() | get the number of elements |

[Example]

The following statement of the following code

The object named second will contain

|  |  |
| --- | --- |
| Method | Return value |
|  | 2 |
|  | 4 |
|  | 3 |

[namespace]

std

[library]

<valarray>

It was declared and defined in the valarray.h header file.

[code]

#include <iostream>

#include <string>

#include <cstddef>

#include <valarray>

using namespace std;

template <class T>

void PrintValarray(valarray<T> va, string name)

{

cout<<name<<":"<<endl;

for(int i=0;i<va.size();i++)

{

cout<<va[i]<<" ";

}

cout<<endl;

}

void PrintSlice(slice slc,string name)

{

cout<<name<<":"<<endl;

cout<<slc.start()<<" "<<slc.stride()<<" "<<slc.size()<<endl;

}

int main ()

{

std::valarray<int> foo (12);

for (int i=0; i<12; ++i)

{

foo[i]=i\*i;

}

std::slice second (2,3,4);

std::slice third (second);

std::valarray<int> bar = foo[third];

PrintValarray<int>(foo,"foo");

cout<<"---------"<<endl;

PrintValarray<int>(bar,"bar");

cout<<"---------"<<endl;

PrintSlice(second,"second");

cout<<"---------"<<endl;

return 0;

}

[result]

foo:

0 1 4 9 16 25 36 49 64 81 100 121

---------

bar:

4 36 100

---------

second:

2 4 3

--------- [ref]

<https://cplusplus.com/reference/valarray/slice/slice/>