library 함수 사용 실습
+ command line arguments

2023

국민대학교 소프트웨어학부

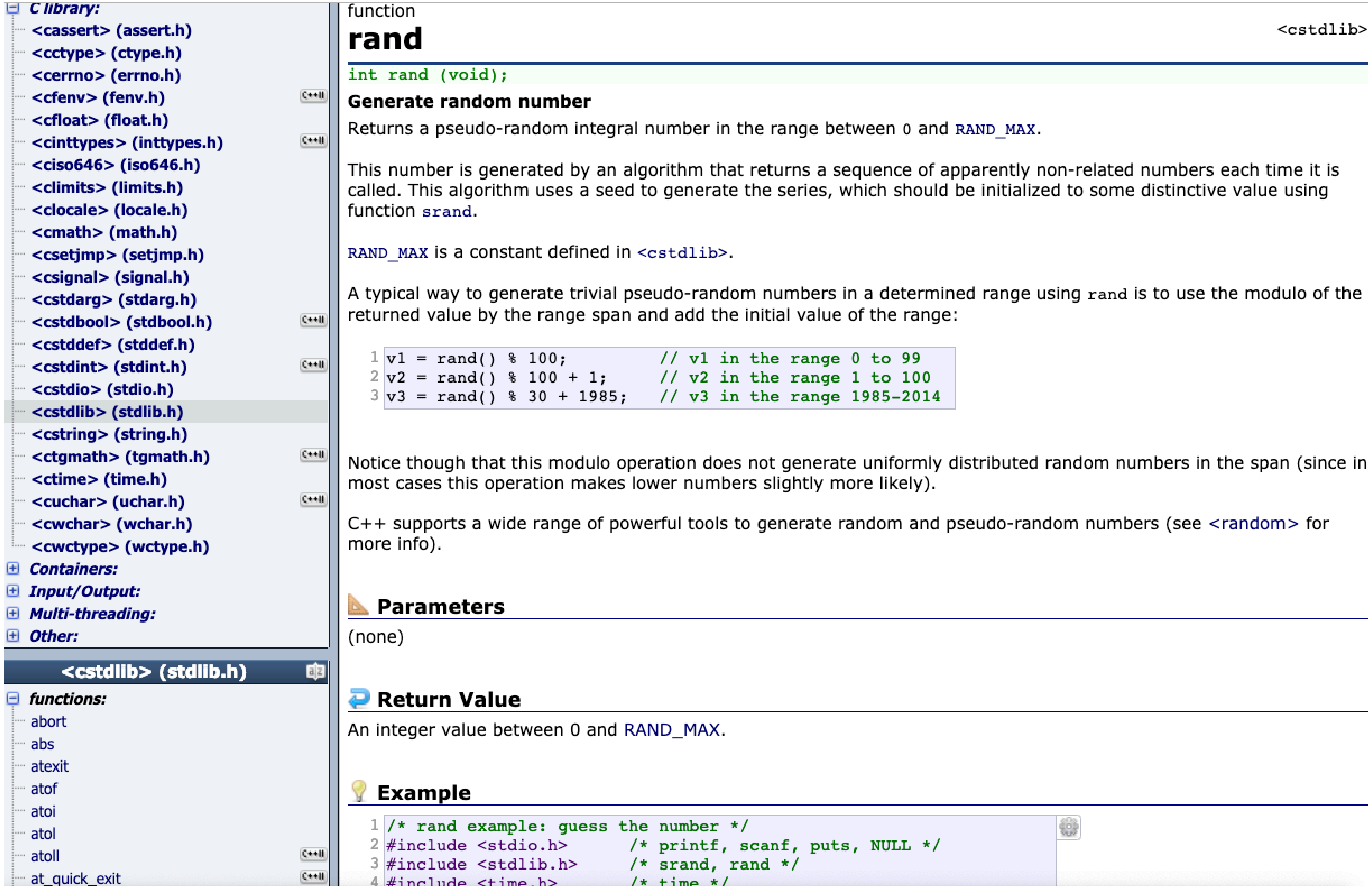
난수 생성 라이브러리 함수

- int rand()
  - 난수를 생성하는 함수
  - 0부터 RAND_MAX까지의 난수를 생성

```
#include <iostream>
#include <cmath>
#include <ctime>
#include <cstdlib>
using namespace std;
// 0에서 RAND_MAX 까지 n개의 난수를 화면에 출력한다.
void get_random( int n )
{
	int i;
	for( i = 0; i < n; i++ )
		cout << rand() << endl;
}
int main()
{
	// 일반적으로 난수 발생기의 시드(seed)를 현재 시간으로 설정한다.
	// 현재 시간은 수행할 때마다 달라지기 때문이다.
	srand( (unsigned)time( NULL ) );
	get_random( 10 );
	return 0;
}
```

```
21783
14153
 4693
13117
21900
19957
15212
20710
 4357
16495
```

search rand at cplusplus.com



C library:

- <cassert> (assert.h)
- <cctype> (ctype.h)
- <cerrno> (errno.h)
- <cfenv> (fenv.h) C++11
- <cfloat> (float.h)
- <cinttypes> (inttypes.h) C++11
- <ciso646> (iso646.h)
- <climits> (limits.h)
- <clocale> (locale.h)
- <cmath> (math.h)
- <csetjmp> (setjmp.h)
- <csignal> (signal.h)
- <cstdarg> (stdarg.h)
- <cstdbool> (stdbool.h) C++11
- <cstddef> (stddef.h)
- <cstdint> (stdint.h) C++11
- <cstdio> (stdio.h)
- <cstdlib> (stdlib.h)
- <cstring> (string.h)
- <ctgmath> (tgmath.h) C++11
- <ctime> (time.h)
- <cuchar> (uchar.h) C++11
- <cwchar> (wchar.h)
- <cwctype> (wctype.h)

Containers:

Input/Output:

Multi-threading:

Other:

<cstdlib> (stdlib.h)

functions:

- abort
- abs
- atexit
- atof
- atoi
- atol
- atoll C++11
- at_quick_exit C++11

function

rand

<cstdlib>

```
int rand (void);
```

Generate random number

Returns a pseudo-random integral number in the range between `0` and `RAND_MAX`.

This number is generated by an algorithm that returns a sequence of apparently non-related numbers each time it is called. This algorithm uses a seed to generate the series, which should be initialized to some distinctive value using function `srand`.

`RAND_MAX` is a constant defined in `<cstdlib>`.

A typical way to generate trivial pseudo-random numbers in a determined range using `rand` is to use the modulo of the returned value by the range span and add the initial value of the range:

```
v1 = rand() % 100;         // v1 in the range 0 to 99
v2 = rand() % 100 + 1;     // v2 in the range 1 to 100
v3 = rand() % 30 + 1985;   // v3 in the range 1985-2014
```

Notice though that this modulo operation does not generate uniformly distributed random numbers in the span (since in most cases this operation makes lower numbers slightly more likely).

C++ supports a wide range of powerful tools to generate random and pseudo-random numbers (see <random> for more info).

Parameters

(none)

Return Value

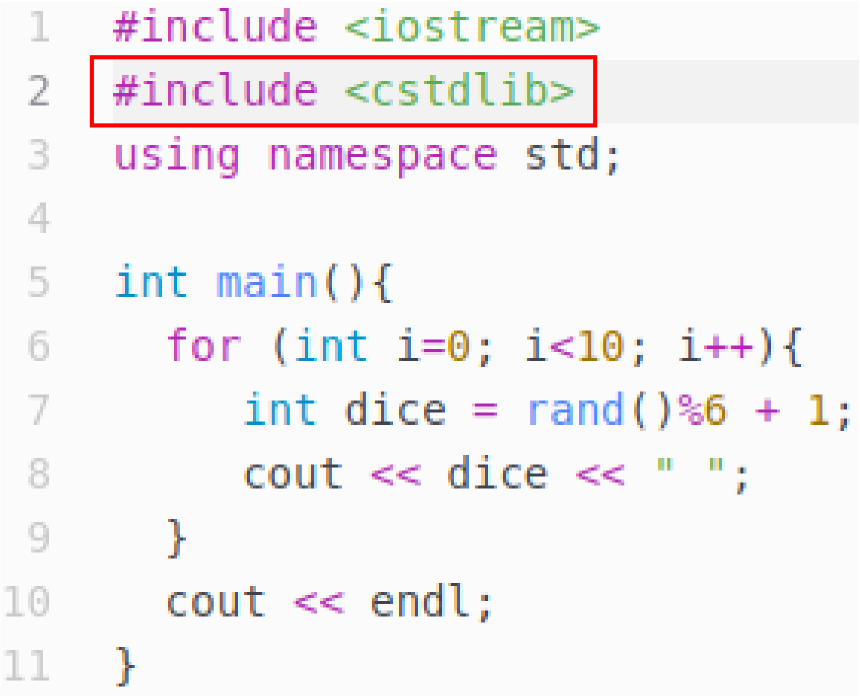
An integer value between 0 and RAND_MAX.

Example

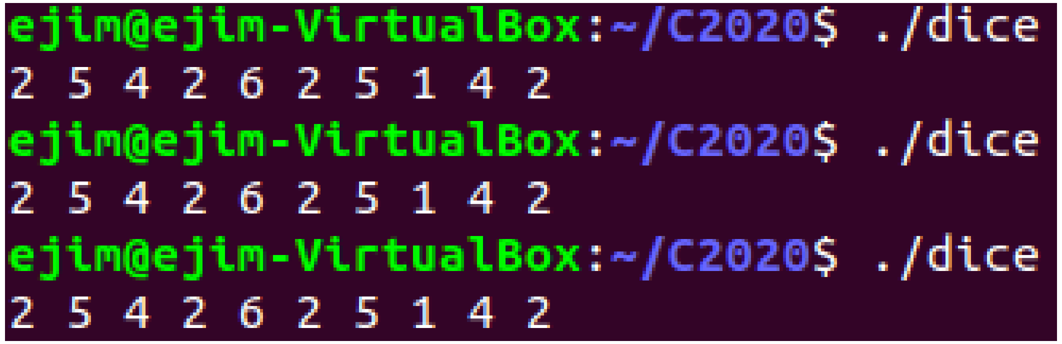
```
/* rand example: guess the number */
#include <stdio.h>      /* printf, scanf, puts, NULL */
#include <stdlib.h>     /* srand, rand */
#include <time.h>       /* time */
```

rand() 함수 사용하여 주사위 던지기 – 틀린 답

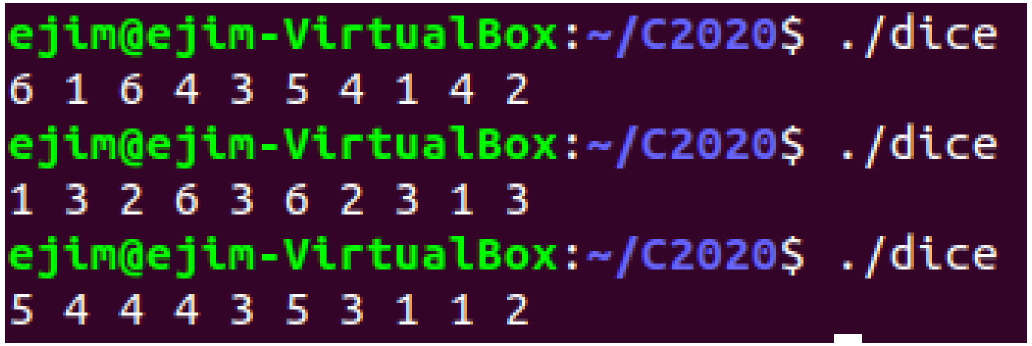
수행할 때마다 1,2,3,4,5,6 중 예측 불가능한 수를 10 개 random 으로 출력하는 프로그램을 작성하라.



```
#include <iostream>
#include <cstdlib>
using namespace std;

int main(){
  for (int i=0; i<10; i++){
      int dice = rand()%6 + 1;
      cout << dice << " ";
  }
  cout << endl;
}
```

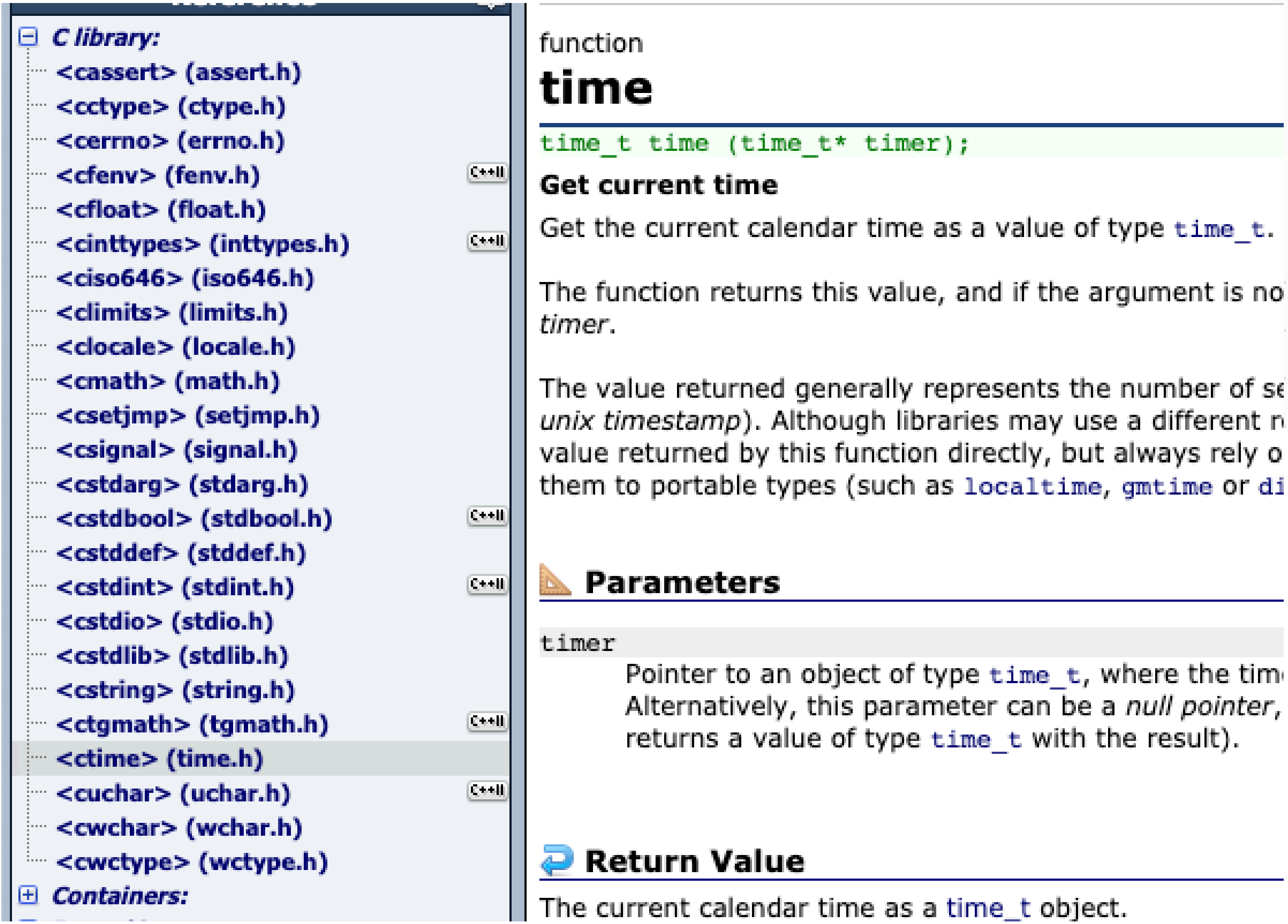


```
ejim@ejim-VirtualBox:~/C2020$ ./dice
2 5 4 2 6 2 5 1 4 2
ejim@ejim-VirtualBox:~/C2020$ ./dice
2 5 4 2 6 2 5 1 4 2
ejim@ejim-VirtualBox:~/C2020$ ./dice
2 5 4 2 6 2 5 1 4 2
```



```
ejim@ejim-VirtualBox:~/C2020$ ./dice
6 1 6 4 3 5 4 1 4 2
ejim@ejim-VirtualBox:~/C2020$ ./dice
1 3 2 6 3 6 2 3 1 3
ejim@ejim-VirtualBox:~/C2020$ ./dice
5 4 4 4 3 5 3 1 1 2
```

time() in <ctime>



C library:

- <cassert> (assert.h)
- <cctype> (ctype.h)
- <cerrno> (errno.h)
- <cfenv> (fenv.h) C++11
- <cfloat> (float.h)
- <cinttypes> (inttypes.h) C++11
- <ciso646> (iso646.h)
- <climits> (limits.h)
- <clocale> (locale.h)
- <cmath> (math.h)
- <csetjmp> (setjmp.h)
- <csignal> (signal.h)
- <cstdarg> (stdarg.h)
- <cstdbool> (stdbool.h) C++11
- <cstddef> (stddef.h)
- <cstdint> (stdint.h) C++11
- <cstdio> (stdio.h)
- <cstdlib> (stdlib.h)
- <cstring> (string.h)
- <ctgmath> (tgmath.h) C++11
- <ctime> (time.h)
- <cuchar> (uchar.h) C++11
- <cwchar> (wchar.h)
- <cwctype> (wctype.h)

Containers:

function

time

```
time_t time (time_t* timer);
```

Get current time

Get the current calendar time as a value of type `time_t`.

The function returns this value, and if the argument is no
timer.

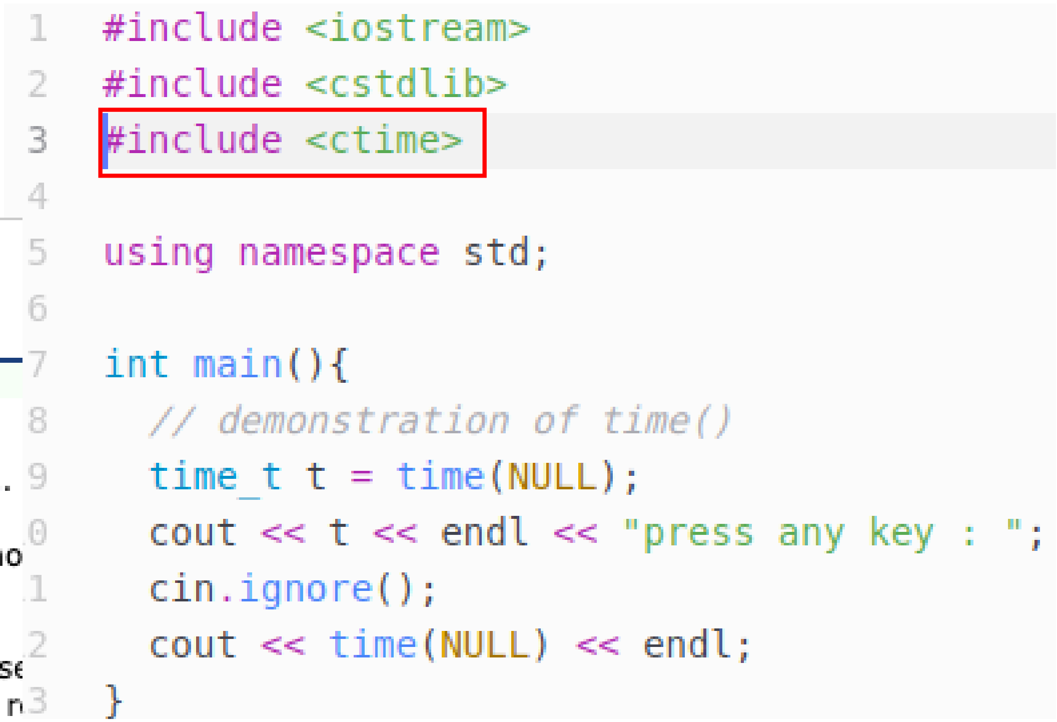
The value returned generally represents the number of se
unix timestamp). Although libraries may use a different r
value returned by this function directly, but always rely o
them to portable types (such as `localtime`, `gmtime` or di

Parameters

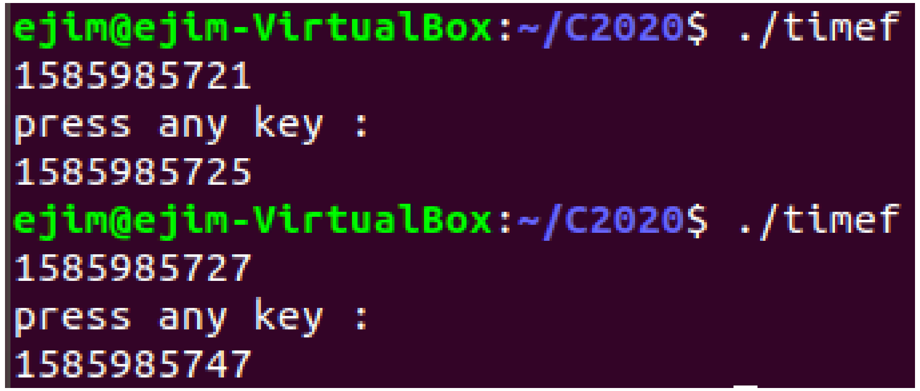
`timer`
Pointer to an object of type `time_t`, where the tim
Alternatively, this parameter can be a *null pointer*,
returns a value of type `time_t` with the result).

Return Value

The current calendar time as a `time_t` object.



```
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main(){
  // demonstration of time()
  time_t t = time(NULL);
  cout << t << endl << "press any key : ";
  cin.ignore();
  cout << time(NULL) << endl;
}
```

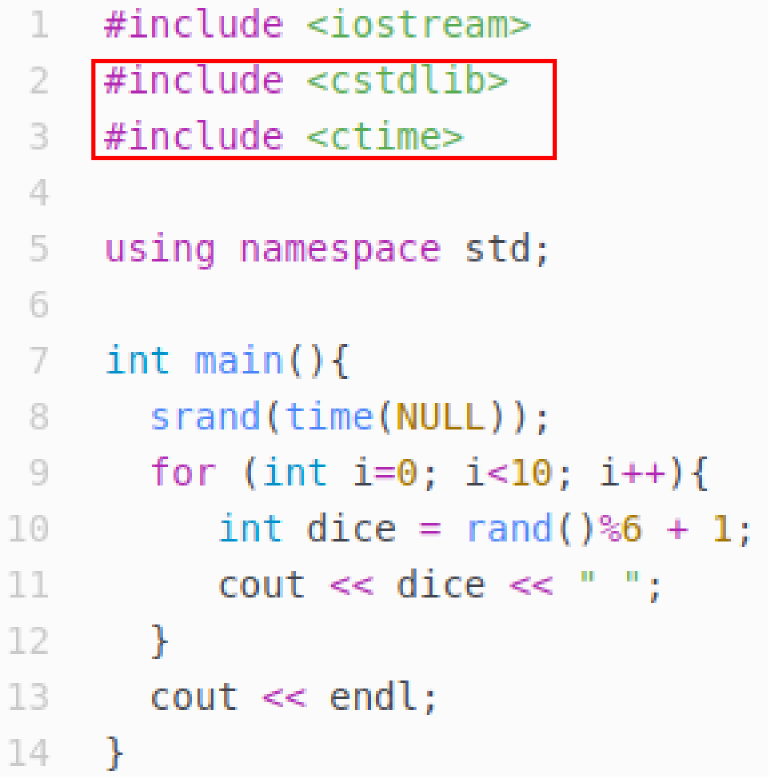


```
ejim@ejim-VirtualBox:~/C2020$ ./timef
1585985721
press any key :
1585985725
ejim@ejim-VirtualBox:~/C2020$ ./timef
1585985727
press any key :
1585985747
```

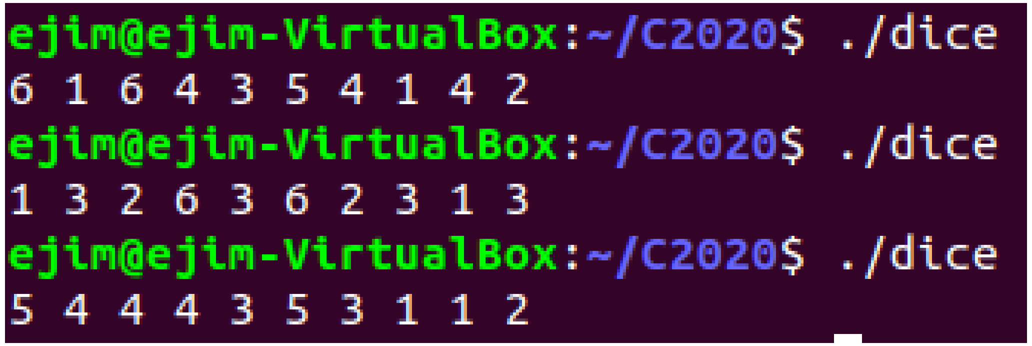
cplusplus.com

rand() 함수 사용하여 주사위 던지기 – srand() 사용

수행할 때마다 1,2,3,4,5,6 중 예측 불가능한 수를 10 개 random 으로 출력하는 프로그램을 작성하라.

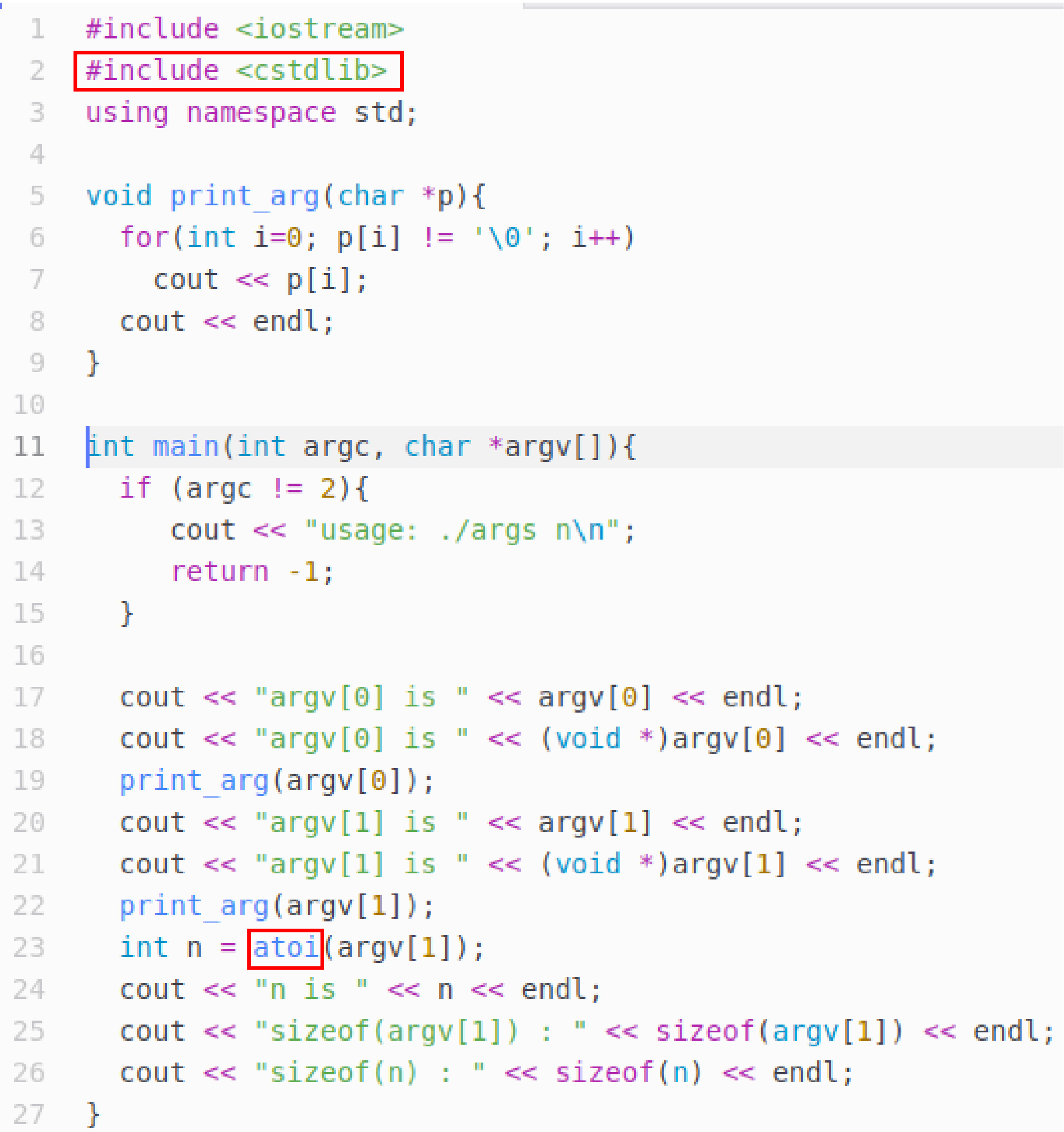


```cpp
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main(){
  srand(time(NULL));
  for (int i=0; i<10; i++){
     int dice = rand()%6 + 1;
     cout << dice << " ";
  }
  cout << endl;
}
```

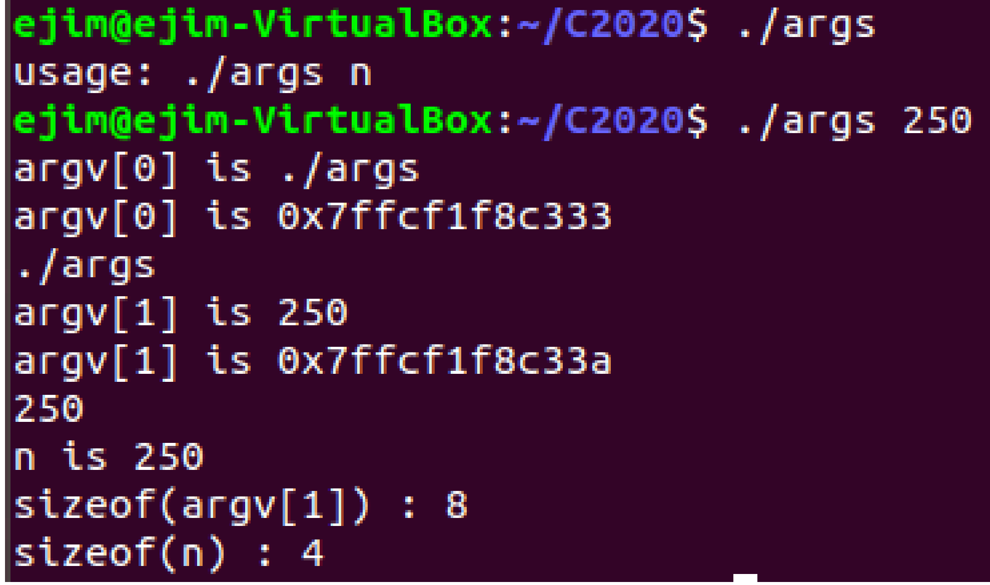


```
ejim@ejim-VirtualBox:~/C2020$ ./dice
6 1 6 4 3 5 4 1 4 2
ejim@ejim-VirtualBox:~/C2020$ ./dice
1 3 2 6 3 6 2 3 1 3
ejim@ejim-VirtualBox:~/C2020$ ./dice
5 4 4 4 3 5 3 1 1 2
```

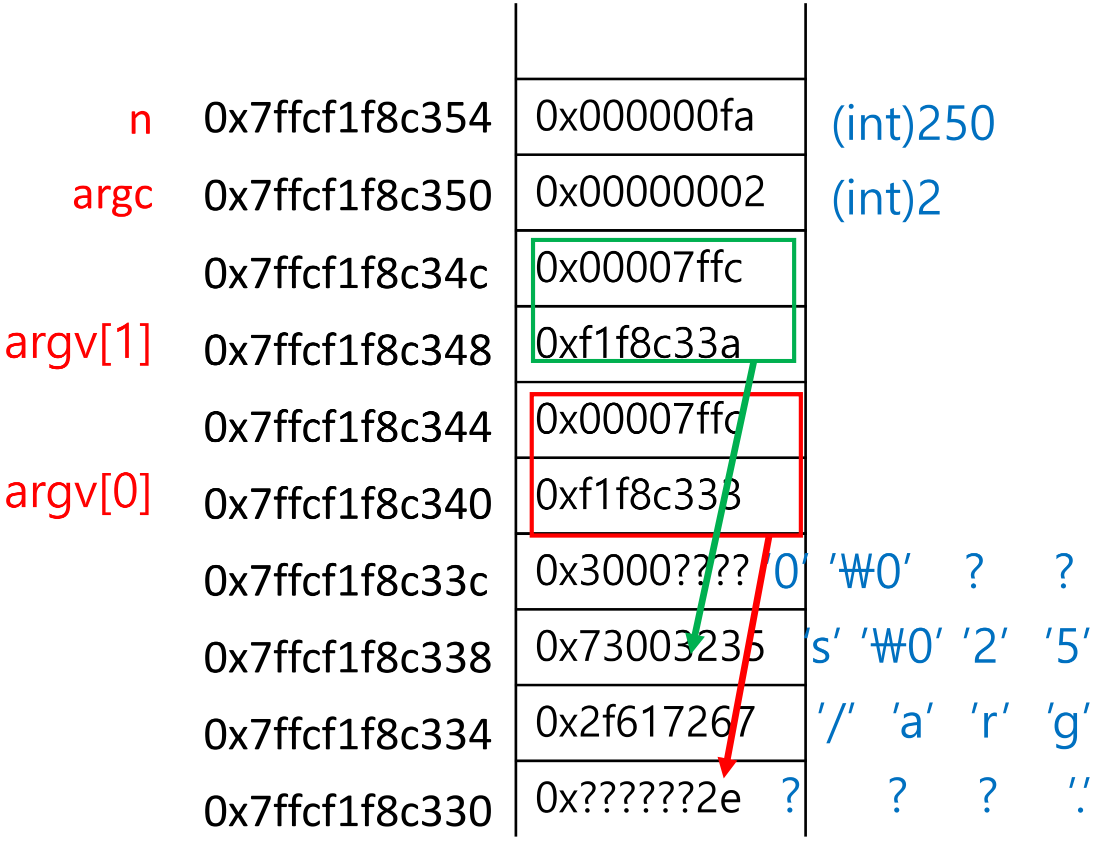
command line arguments : argc and argv



```cpp
#include <iostream>
#include <cstdlib>
using namespace std;

void print_arg(char *p){
  for(int i=0; p[i] != '\0'; i++)
    cout << p[i];
  cout << endl;
}

int main(int argc, char *argv[]){
  if (argc != 2){
     cout << "usage: ./args n\n";
     return -1;
  }

  cout << "argv[0] is " << argv[0] << endl;
  cout << "argv[0] is " << (void *)argv[0] << endl;
  print_arg(argv[0]);
  cout << "argv[1] is " << argv[1] << endl;
  cout << "argv[1] is " << (void *)argv[1] << endl;
  print_arg(argv[1]);
  int n = atoi(argv[1]);
  cout << "n is " << n << endl;
  cout << "sizeof(argv[1]) : " << sizeof(argv[1]) << endl;
  cout << "sizeof(n) : " << sizeof(n) << endl;
}
```

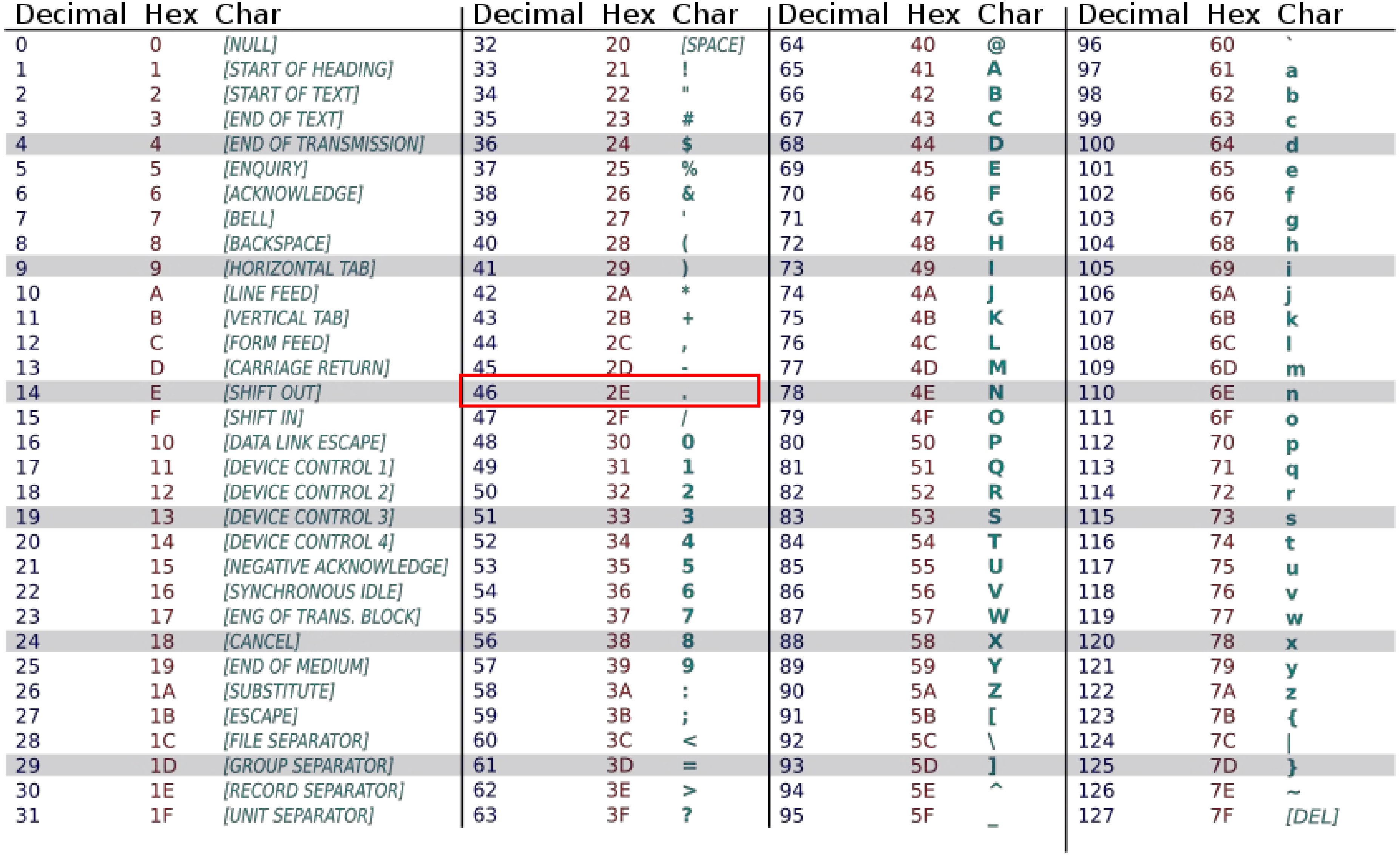


```
ejim@ejim-VirtualBox:~/C2020$ ./args
usage: ./args n
ejim@ejim-VirtualBox:~/C2020$ ./args 250
argv[0] is ./args
argv[0] is 0x7ffcf1f8c333
./args
argv[1] is 250
argv[1] is 0x7ffcf1f8c33a
250
n is 250
sizeof(argv[1]) : 8
sizeof(n) : 4
```



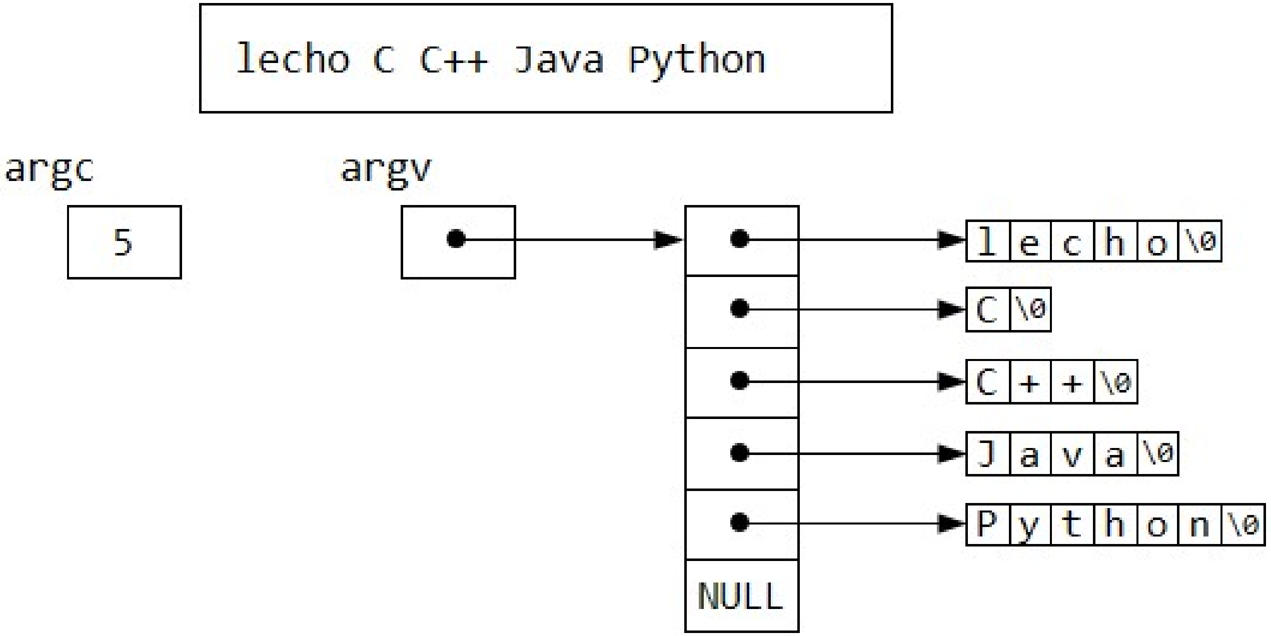
n	0x7ffcf1f8c354	0x000000fa	(int)250
argc	0x7ffcf1f8c350	0x00000002	(int)2
	0x7ffcf1f8c34c	0x00007ffc	
argv[1]	0x7ffcf1f8c348	0xf1f8c33a	
	0x7ffcf1f8c344	0x00007ffc	
argv[0]	0x7ffcf1f8c340	0xf1f8c333	
	0x7ffcf1f8c33c	0x3000????	‘0’ ‘₩0’ ? ?
	0x7ffcf1f8c338	0x73003235	‘s’ ‘₩0’ ‘2’ ‘5’
	0x7ffcf1f8c334	0x2f617267	‘/’ ‘a’ ‘r’ ‘g’
	0x7ffcf1f8c330	0x??????2e	? ? ? ‘.’

ASCII TABLE



Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

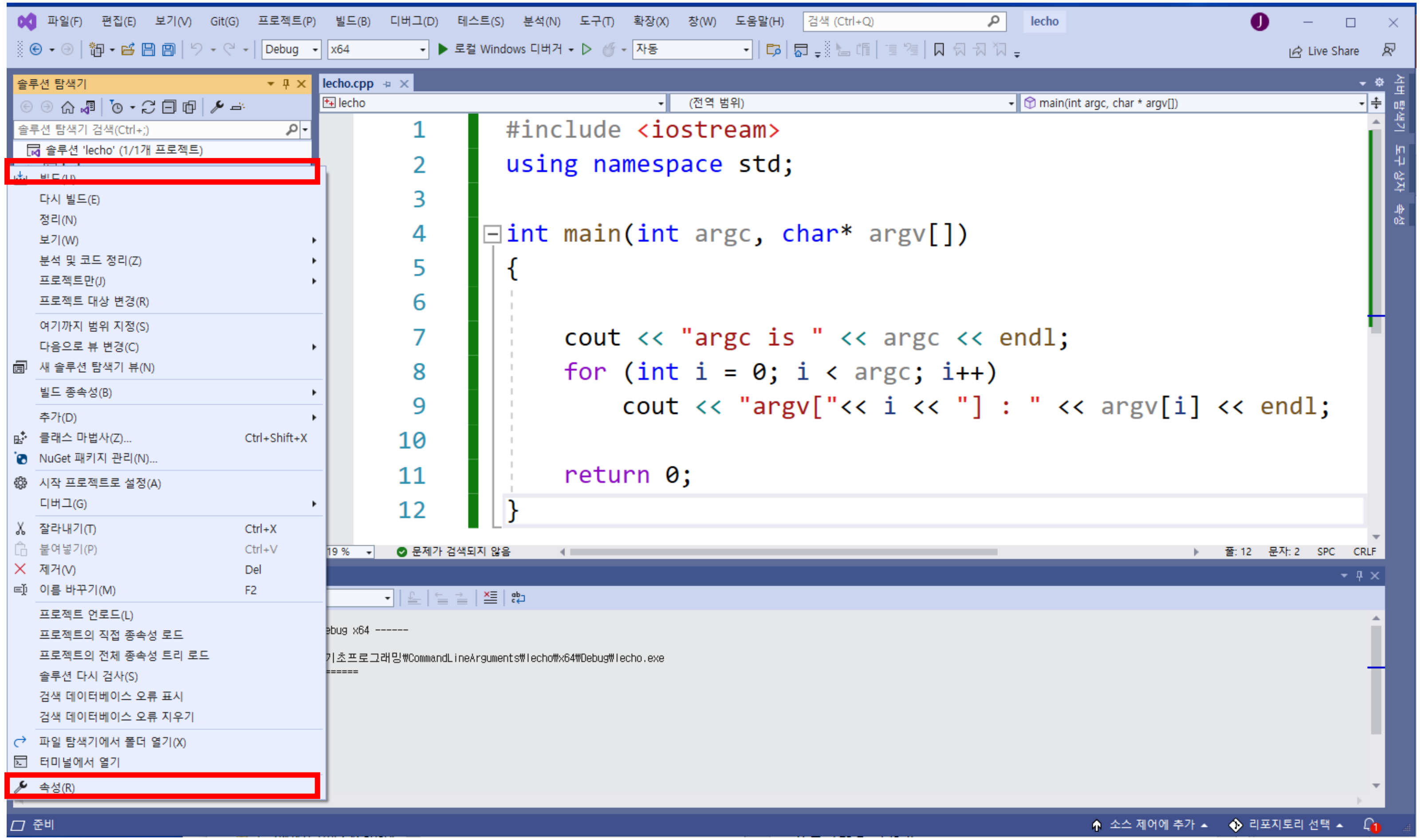
command line arguments : argc and argv



command line arguments : argc and argv

MS Visual Studio에서 command line arguments(명령인수) 입력

<메뉴> : project (마우스 우클릭) -> 속성

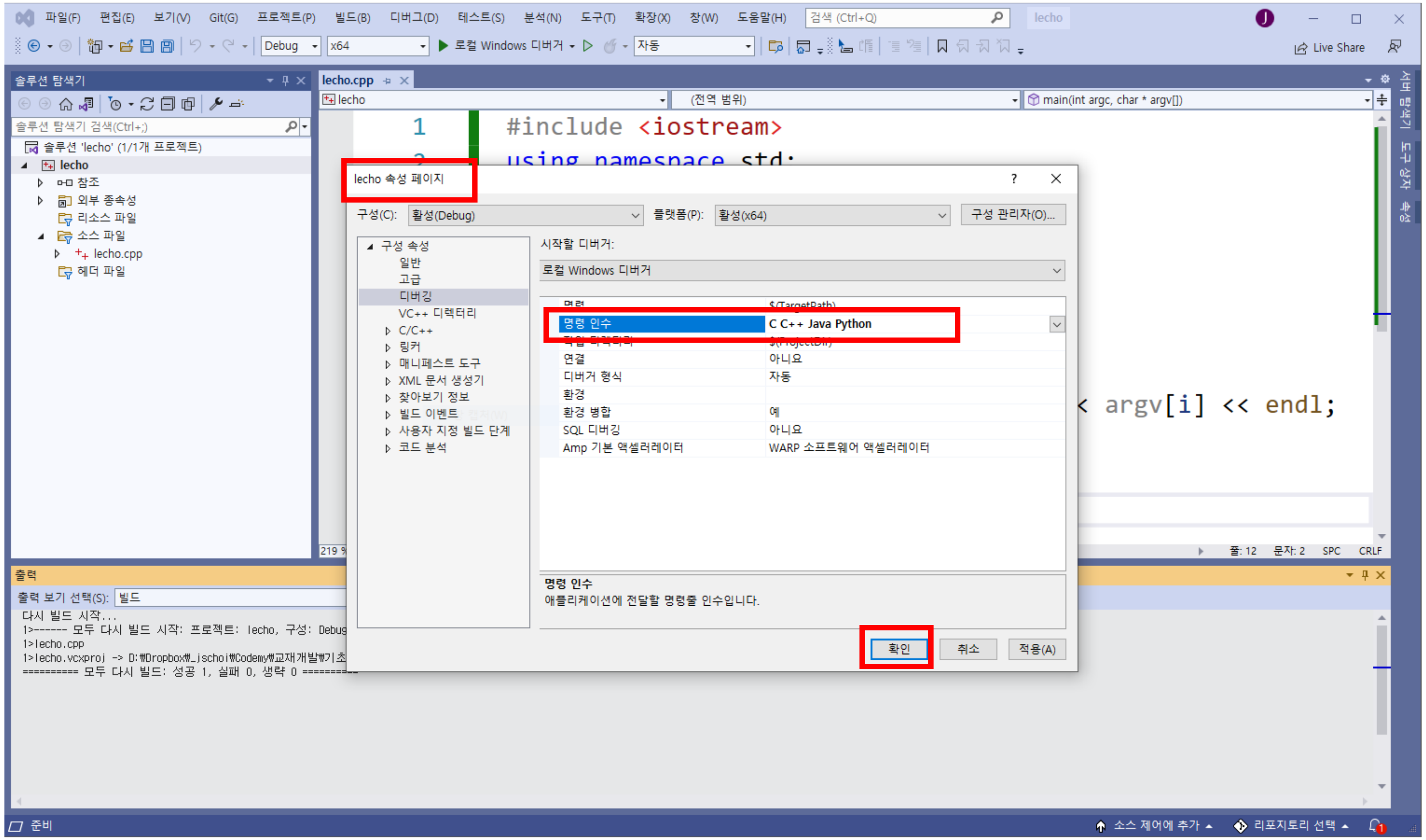


```cpp
#include <iostream>
using namespace std;

int main(int argc, char* argv[])
{

    cout << "argc is " << argc << endl;
    for (int i = 0; i < argc; i++)
        cout << "argv["<< i << "] : " << argv[i] << endl;

    return 0;
}
```

```
ebug x64 ------

기초프로그래밍₩CommandLineArguments₩lecho₩x64₩Debug₩lecho.exe
======
```

command line arguments : argc and argv

<속성 페이지> -> 명령인수 메뉴에서 입력



```
1   #include <iostream>
2   using namespace std;
...
< argv[i] << endl;
```

lecho 속성 페이지

구성(C): 활성(Debug)   플랫폼(P): 활성(x64)   구성 관리자(O)...

시작할 디버거: 로컬 Windows 디버거

명령	$(TargetPath)
명령 인수	C C++ Java Python
작업 디렉터리	$(ProjectDir)
연결	아니요
디버거 형식	자동
환경	
환경 병합	예
SQL 디버깅	아니요
Amp 기본 액셀러레이터	WARP 소프트웨어 액셀러레이터

명령 인수
애플리케이션에 전달할 명령줄 인수입니다.

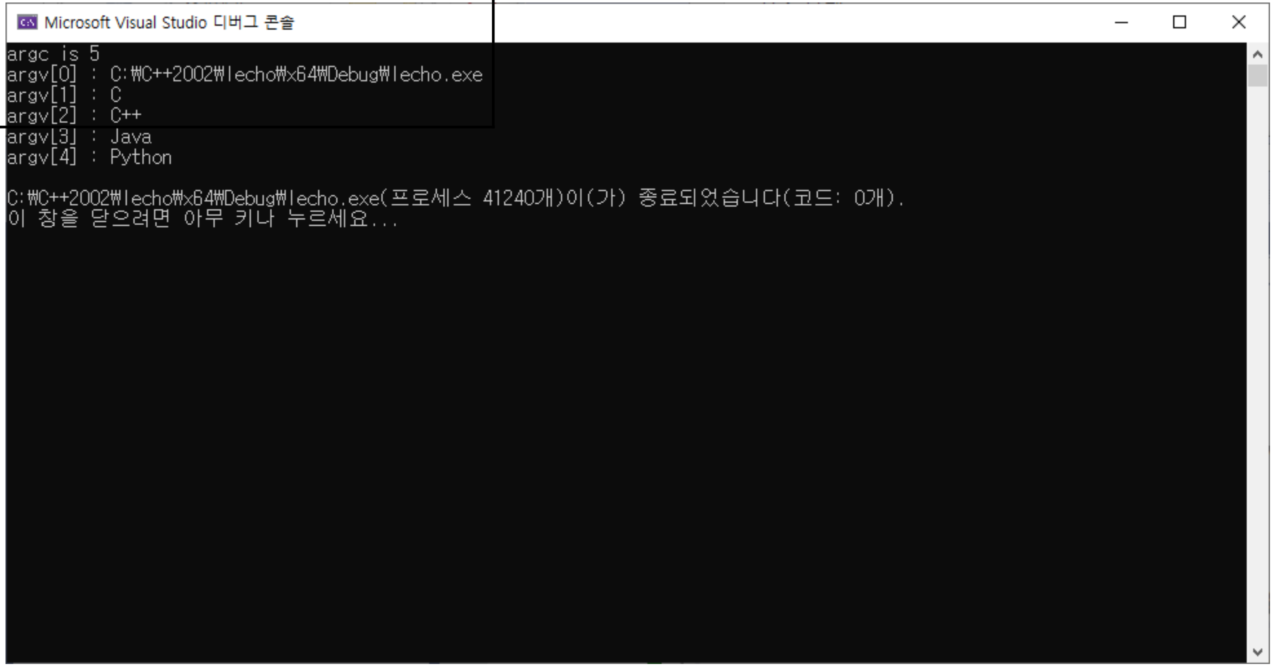
확인 | 취소 | 적용(A)

출력
출력 보기 선택(S): 빌드

```
다시 빌드 시작...
1>------ 모두 다시 빌드 시작: 프로젝트: lecho, 구성: Debug
1>lecho.cpp
1>lecho.vcxproj -> D:₩Dropbox₩_jschoi₩Codemy₩교재개발₩기초
========== 모두 다시 빌드: 성공 1, 실패 0, 생략 0 ==========
```

command line arguments : argc and argv

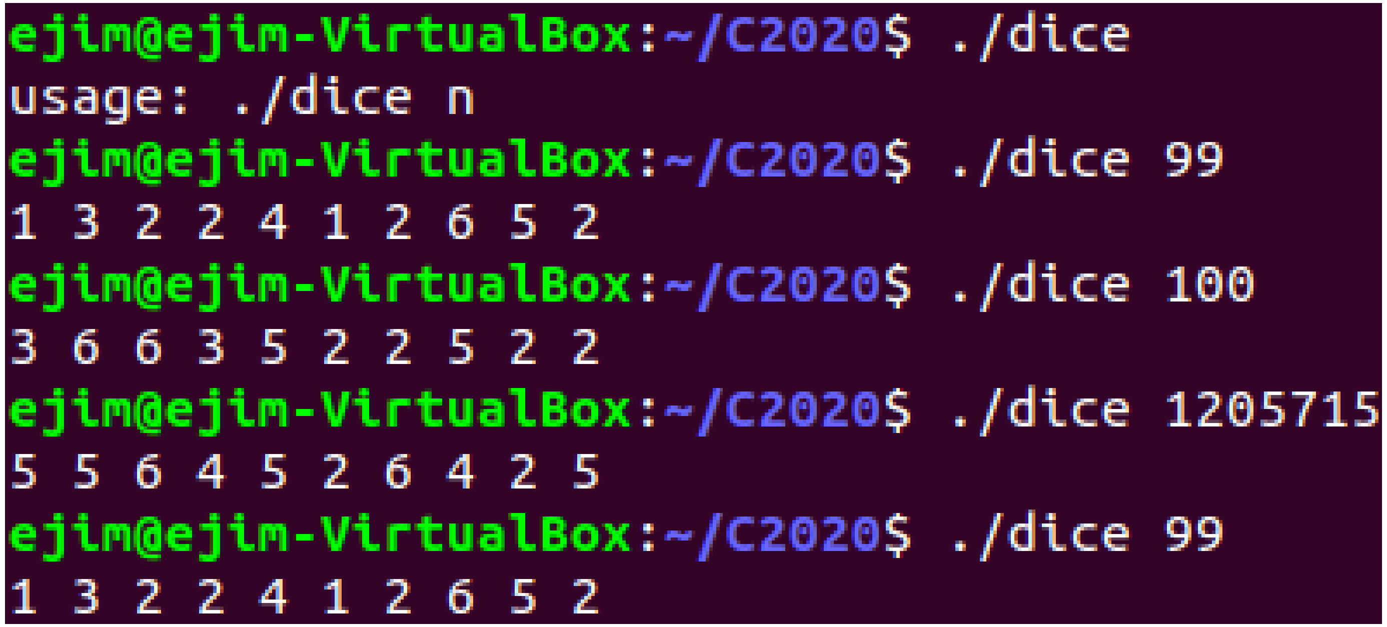
```
#include <iostream>
using namespace std;

int main(int argc, char* argv[])
{

    cout << "argc is " << argc << endl;
    for (int i = 0; i < argc; i++)
        cout << "argv[" << i << "] : " << argv[i] << endl;

    return 0;
}
```



```
argc is 5
argv[0] : C:₩C++2002₩lecho₩x64₩Debug₩lecho.exe
argv[1] : C
argv[2] : C++
argv[3] : Java
argv[4] : Python

C:₩C++2002₩lecho₩x64₩Debug₩lecho.exe(프로세스 41240개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

실습 문제:

command line argument (argv) n 을 사용하여
n 의 값이 같으면 같은 sequence 를 반복하도록 앞의
프로그램을 고쳐라.



```
ejim@ejim-VirtualBox:~/C2020$ ./dice
usage: ./dice n
ejim@ejim-VirtualBox:~/C2020$ ./dice 99
1 3 2 2 4 1 2 6 5 2
ejim@ejim-VirtualBox:~/C2020$ ./dice 100
3 6 6 3 5 2 2 5 2 2
ejim@ejim-VirtualBox:~/C2020$ ./dice 1205715
5 5 6 4 5 2 6 4 2 5
ejim@ejim-VirtualBox:~/C2020$ ./dice 99
1 3 2 2 4 1 2 6 5 2
```