Task

Take date as input in the following format.

```
01042016
```

Task

Take date as input in the following format.

```
01042016
```

Task

Take date as input in the following format.

```
01042016
```

Task

Take date as input in the following format.

```
01042016
```

Task

Take date as input in the following format.

```
01042016
```

Solution

```
    scanf("%2d%2d%4d", &day, &month, &year);
    printf("%d %d %d\n", day, month, year);
```

Task

Take date as input in the following format.

```
01042016
```

Solution

```
scanf("%2d%2d%4d", &day, &month, &year);
printf("%d %d %d\n", day, month, year);
```

Read 2 characters from input as integer.

Task

Take date as input in the following format.

```
01042016
```

Solution

```
scanf("%2d%2d%4d", &day, &month, &year);
printf("%d %d %d\n", day, month, year);
```

Read 2 characters from input as integer.

Task

Take date as input in the following format.

```
01042016
```

Solution

```
scanf("%2d%2d%4d", &day, &month, &year);
printf("%d %d %d\n", day, month, year);
```

Read 4 characters from input as integer.

Task

Take date as input in the following format.

```
01042016
```

Solution

```
scanf("%2d%2d%4d", &day, &month, &year);
printf("%d %d %d\n", day, month, year);
```

```
$
```

Task

Take date as input in the following format.

01042016

Solution

```
scanf("%2d%2d%4d", &day, &month, &year);
printf("%d %d %d\n", day, month, year);
```

```
$ ./a.out ↵
```

Task

Take date as input in the following format.

01042016

Solution

```c
    scanf("%2d%2d%4d", &day, &month, &year);
    printf("%d %d %d\n", day, month, year);
```

```
$ ./a.out ↵
31032016 ↵
```

Task

Take date as input in the following format.

```
01042016
```

Solution

```
scanf("%2d%2d%4d", &day, &month, &year);
printf("%d %d %d\n", day, month, year);
```

```
$ ./a.out ↵
31032016 ↵
31 3 2016 ↵
$
```

Task

Read percentages from input.

50%

Task

Read percentages from input.

$$50\%$$

Solution

```
scanf("%d%%", &a);
printf("%d\n", a);
```

Task

Read percentages from input.

50%

Solution

```
    scanf("%d%%", &a);
    printf("%d\n", a);
```

Task

Read percentages from input.

50%

Solution

```
    scanf("%d%%", &a);
    printf("%d\n", a);
```

Task

Read percentages from input.

50%

Solution

```
scanf("%d%%", &a);
printf("%d\n", a);
```

```
$
```

Task

Read percentages from input.

50%

Solution

```
scanf("%d%%", &a);
printf("%d\n", a);
```

```
$ ./a.out ↵
```

Task

Read percentages from input.

50%

Solution

```
scanf("%d%%", &a);
printf("%d\n", a);
```

```
$ ./a.out ↵
200% ↵
```

Task

Read percentages from input.

$$50\%$$

Solution

```
scanf("%d%%", &a);
printf("%d\n", a);
```

```
$ ./a.out ↵
200% ↵
200
$
```

Format String	Input	Output

Format String	Input	Output
`"%3f%2d"`		

Format String	Input	Output
`"%3f%2d"`	`123456`	

Format String	Input	Output
`"%3f%2d"`	`123456`	`123.000000 45`

Format String	Input	Output
"%3f%2d"	123456	123.000000 45
"%3f %2d"		

Format String	Input	Output
`"%3f%2d"`	`123456`	`123.000000 45`
`"%3f %2d"`	`123456`	

Format String	Input	Output
`"%3f%2d"`	`123456`	`123.000000 45`
`"%3f %2d"`	`123456`	`123.000000 45`

Format String	Input	Output
`"%3f%2d"`	`123456`	`123.000000 45`
`"%3f %2d"`	`123456`	`123.000000 45`
`"%3s"`		

Format String	Input	Output
`"%3f%2d"`	`123456`	`123.000000 45`
`"%3f %2d"`	`123456`	`123.000000 45`
`"%3s"`	`123456`	

Format String	Input	Output
`"%3f%2d"`	`123456`	`123.000000 45`
`"%3f %2d"`	`123456`	`123.000000 45`
`"%3s"`	`123456`	`123`

Format String	Input	Output
`"%3f%2d"`	`123456`	`123.000000 45`
`"%3f %2d"`	`123456`	`123.000000 45`
`"%3s"`	`123456`	`123`
`"%3d%3s %c%f"`		

Format String	Input	Output
`"%3f%2d"`	`123456`	`123.000000 45`
`"%3f %2d"`	`123456`	`123.000000 45`
`"%3s"`	`123456`	`123`
`"%3d%3s %c%f"`	`123 456 789 012`	

Format String	Input	Output
`"%3f%2d"`	`123456`	`123.000000 45`
`"%3f %2d"`	`123456`	`123.000000 45`
`"%3s"`	`123456`	`123`
`"%3d%3s %c%f"`	`123 456 789 012`	`123 456 7 89.000000`

```
$
```

```
char *str = "hello, world";
printf(":%s:\n", str);
```

```
$
```

```c
char *str = "hello, world";
printf(":%s:\n", str);
```

```
$ ./a.out ↵
```

```c
char *str = "hello, world";
printf(":%s:\n", str);
```

```
$ ./a.out ↵
:hello,␣world:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
```

```
$ ./a.out ↵
:hello,␣world:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
```

number – min. width

```
$ ./a.out ↵
:hello,␣world:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
```

number – min. width

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
```

number – min. width

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
```

number – min. width

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
:hello,␣world:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
printf(":%-15s:\n", str);
```

number – min. width

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
:hello,␣world:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
printf(":%-15s:\n", str);
```

number – min. width

minus – adjust left

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
:hello,␣world:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
printf(":%-15s:\n", str);
```

number – min. width

minus – adjust left

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
:hello,␣world:
:hello,␣world␣␣␣:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
printf(":%-15s:\n", str);
printf(":%.10s:\n", str);
```

number – min. width

minus – adjust left

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
:hello,␣world:
:hello,␣world␣␣␣:
```

```c
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
printf(":%-15s:\n", str);
printf(":%.10s:\n", str);
```

number – min. width

minus – adjust left

.number – max. characters

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
:hello,␣world:
:hello,␣world␣␣␣:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
printf(":%-15s:\n", str);
printf(":%.10s:\n", str);
```

number – min. width

minus – adjust left

.number – max. characters

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
:hello,␣world:
:hello,␣world␣␣␣:
:hello,␣wor:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
printf(":%-15s:\n", str);
printf(":%.10s:\n", str);
printf(":%15.10s:\n", str);
```

number – min. width

minus – adjust left

.number – max. characters

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
:hello,␣world:
:hello,␣world␣␣␣:
:hello,␣wor:
```

```c
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
printf(":%-15s:\n", str);
printf(":%.10s:\n", str);
printf(":%15.10s:\n", str);
```

number – min. width

minus – adjust left

.number – max. characters

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
:hello,␣world:
:hello,␣world␣␣␣:
:hello,␣wor:
:␣␣␣␣␣hello,␣wor:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
printf(":%-15s:\n", str);
printf(":%.10s:\n", str);
printf(":%15.10s:\n", str);
printf(":%-15.10s:\n", str);
```

number – min. width

minus – adjust left

.number – max. characters

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
:hello,␣world:
:hello,␣world␣␣␣:
:hello,␣wor:
:␣␣␣␣␣hello,␣wor:
```

```
char *str = "hello, world";
printf(":%s:\n", str);
printf(":%15s:\n", str);
printf(":%10s:\n", str);
printf(":%-15s:\n", str);
printf(":%.10s:\n", str);
printf(":%15.10s:\n", str);
printf(":%-15.10s:\n", str);
```

number – min. width

minus – adjust left

.number – max. characters

```
$ ./a.out ↵
:hello,␣world:
:␣␣␣hello,␣world:
:hello,␣world:
:hello,␣world␣␣␣:
:hello,␣wor:
:␣␣␣␣␣hello,␣wor:
:hello,␣wor␣␣␣␣␣:
```

```
$
```

```
int a = 12;
printf(":%d:\n", a);
```

```
$
```

```
int a = 12;
printf(":%d:\n", a);
```

```
$ ./a.out ↵
```

```
int a = 12;
printf(":%d:\n", a);
```

```
$ ./a.out ↵
:12:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
```

```
$ ./a.out ↵
:12:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
```

number – min. width

```
$ ./a.out ↵
:12:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
```

number – min. width

```
$ ./a.out ↵
:12:
:␣12:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
```

number – min. width

```
$ ./a.out ↵
:12:
:␣12:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
```

number – min. width

```
$ ./a.out ↵
:12:
:␣12:
:12:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
```

number – min. width

```
$ ./a.out ↵
:12:
:␣12:
:12:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
```

number – min. width

minus – adjust left

```
$ ./a.out ↵
:12:
:␣12:
:12:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
```

number – min. width

minus – adjust left

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
printf(":%.3d:\n", a);
```

number – min. width

minus – adjust left

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
printf(":%.3d:\n", a);
```

number – min. width

minus – adjust left

.number – min. digits, pad 0

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
printf(":%.3d:\n", a);
```

number – min. width

minus – adjust left

.number – min. digits, pad 0

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
:012:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
printf(":%.3d:\n", a);
printf(":%.4d:\n", a);
```

number – min. width

minus – adjust left

.number – min. digits, pad 0

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
:012:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
printf(":%.3d:\n", a);
printf(":%.4d:\n", a);
```

number – min. width

minus – adjust left

.number – min. digits, pad 0

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
:012:
:0012:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
printf(":%.3d:\n", a);
printf(":%.4d:\n", a);
printf(":%-4d:\n", a);
```

number – min. width

minus – adjust left

.number – min. digits, pad 0

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
:012:
:0012:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
printf(":%.3d:\n", a);
printf(":%.4d:\n", a);
printf(":%-4d:\n", a);
```

number – min. width

minus – adjust left

.number – min. digits, pad 0

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
:012:
:0012:
:12␣␣:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
printf(":%.3d:\n", a);
printf(":%.4d:\n", a);
printf(":%-4d:\n", a);
printf(":%4.3d:\n", a);
```

number – min. width

minus – adjust left

.number – min. digits, pad 0

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
:012:
:0012:
:12␣␣:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
printf(":%.3d:\n", a);
printf(":%.4d:\n", a);
printf(":%-4d:\n", a);
printf(":%4.3d:\n", a);
```

number – min. width

minus – adjust left

.number – min. digits, pad 0

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
:012:
:0012:
:12␣␣:
:␣012:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
printf(":%.3d:\n", a);
printf(":%.4d:\n", a);
printf(":%-4d:\n", a);
printf(":%4.3d:\n", a);
printf(":%-4.3d:\n", a);
```

number – min. width

minus – adjust left

.number – min. digits, pad 0

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
:012:
:0012:
:12␣␣:
:␣012:
```

```
int a = 12;
printf(":%d:\n", a);
printf(":%3d:\n", a);
printf(":%1d:\n", a);
printf(":%-3d:\n", a);
printf(":%.3d:\n", a);
printf(":%.4d:\n", a);
printf(":%-4d:\n", a);
printf(":%4.3d:\n", a);
printf(":%-4.3d:\n", a);
```

number – min. width

minus – adjust left

.number – min. digits, pad 0

```
$ ./a.out ↵
:12:
:␣12:
:12:
:12␣:
:012:
:0012:
:12␣␣:
:␣012:
:012 :
```

```
$
```

```
float f = 12.4356;
printf(":%f:\n", f);
```

```
$
```

```
float f = 12.4356;
printf(":%f:\n", f);
```

```
$ ./a.out ↵
```

```
float f = 12.4356;
printf(":%f:\n", f);
```

```
$ ./a.out ↵
:12.435600:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
```

```
$ ./a.out ↵
:12.435600:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
```

number – min. width

```
$ ./a.out ↵
:12.435600:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
```

number – min. width

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
```

number – min. width

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
```

number – min. width

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
```

number – min. width

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
```

number – min. width

minus – adjust left

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
```

number – min. width

minus – adjust left

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
:12.435600␣:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
printf(":%.3f:\n", f);
```

number – min. width

minus – adjust left

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
:12.435600␣:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
printf(":%.3f:\n", f);
```

number – min. width

minus – adjust left

.number – max. decimal digits

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
:12.435600␣:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
printf(":%.3f:\n", f);
```

number – min. width

minus – adjust left

.number – max. decimal digits

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
:12.435600␣:
:12.436:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
printf(":%.3f:\n", f);
printf(":%-4f:\n", f);
```

number – min. width

minus – adjust left

.number – max. decimal digits

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
:12.435600␣:
:12.436:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
printf(":%.3f:\n", f);
printf(":%-4f:\n", f);
```

number – min. width

minus – adjust left

.number – max. decimal digits

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
:12.435600␣:
:12.436:
:12.4356:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
printf(":%.3f:\n", f);
printf(":%-4f:\n", f);
printf(":%10.3f:\n", f);
```

number – min. width

minus – adjust left

.number – max. decimal digits

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
:12.435600␣:
:12.436:
:12.4356:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
printf(":%.3f:\n", f);
printf(":%-4f:\n", f);
printf(":%10.3f:\n", f);
```

number – min. width

minus – adjust left

.number – max. decimal digits

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
:12.435600␣:
:12.436:
:12.4356:
:␣␣␣␣12.436:
```

```c
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
printf(":%.3f:\n", f);
printf(":%-4f:\n", f);
printf(":%10.3f:\n", f);
printf(":%-10.3f:\n", f);
```

number – min. width

minus – adjust left

.number – max. decimal digits

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
:12.435600␣:
:12.436:
:12.4356:
:␣␣␣␣12.436:
```

```
float f = 12.4356;
printf(":%f:\n", f);
printf(":%10f:\n", f);
printf(":%5f:\n", f);
printf(":%-10f:\n", f);
printf(":%.3f:\n", f);
printf(":%-4f:\n", f);
printf(":%10.3f:\n", f);
printf(":%-10.3f:\n", f);
```

number – min. width

minus – adjust left

.number – max. decimal digits

```
$ ./a.out ↵
:12.435600:
:␣12.435600:
:12.435600:
:12.435600␣:
:12.436:
:12.4356:
:␣␣␣␣12.436:
:12.436␣␣␣␣:
```

```
char *str = "hello, world";
```

```
$
```

```
char *str = "hello, world";
printf(":%*.*s:\n", 10, 3, str);
```

```
$
```

```
char *str = "hello, world";
printf(":%*.*s:\n", 10, 3, str);
```

```
$
```

```
char *str = "hello, world";
printf(":%*.*s:\n", 10, 3, str);
```

```
$ ./a.out ↵
```

```
char *str = "hello, world";
printf(":%*.*s:\n", 10, 3, str);
```

```
$ ./a.out ↵
:␣␣␣␣␣␣␣hel:
```

```
char *str = "hello, world";
```

```
printf(":%*.*s:\n", 10, 3, str);
printf(":%-*.*s:\n", 10, 3, str);
```

```
$ ./a.out ↵
:␣␣␣␣␣␣␣hel:
```

```
char *str = "hello, world";

printf(":%*.*s:\n", 10, 3, str);
printf(":%-*.*s:\n", 10, 3, str);
```

```
$ ./a.out ↵
:␣␣␣␣␣␣␣hel:
:hel␣␣␣␣␣␣␣:
```

```
char *str = "hello, world";

printf(":%*.*s:\n", 10, 3, str);
printf(":%-*.*s:\n", 10, 3, str);

int a = 10, b = 3;
```

```
$ ./a.out ↵
:␣␣␣␣␣␣␣hel:
:hel␣␣␣␣␣␣␣:
```

```
char *str = "hello, world";

printf(":%*.*s:\n", 10, 3, str);
printf(":%-*.*s:\n", 10, 3, str);

int a = 10, b = 3;
printf(":%*.*s:\n", a, b, str);
```

```
$ ./a.out ↵
:␣␣␣␣␣␣␣hel:
:hel␣␣␣␣␣␣␣:
```

```c
char *str = "hello, world";

printf(":%*.*s:\n", 10, 3, str);
printf(":%-*.*s:\n", 10, 3, str);

int a = 10, b = 3;
printf(":%*.*s:\n", a, b, str);
```

```
$ ./a.out ↵
:␣␣␣␣␣␣␣hel:
:hel␣␣␣␣␣␣␣:
:␣␣␣␣␣␣␣hel:
```

```c
char *str = "hello, world\n";
```

```
$
```

```
char *str = "hello, world\n";
printf(str);
```

```
$
```

```
char *str = "hello, world\n";
printf(str);
```

```
$ ./a.out ↵
```

```
char *str = "hello, world\n";
printf(str);
```

```
$ ./a.out ↵
hello, world
```

```
char *str = "hello, world\n";
printf(str);
```

- Format string need not be fixed.

```
$ ./a.out ↵
hello, world
```

```
char *str = "hello, world %d\n";
printf(str);
```

- Format string need not be fixed.

```
$ ./a.out ↵
hello, world
```

```
char *str = "hello, world %d\n";
printf(str, 10);
```

- Format string need not be fixed.

```
$ ./a.out ↵
hello, world
```

```
char *str = "hello, world %d\n";
printf(str, 10);
```

- Format string need not be fixed.

```
$ ./a.out ↵
hello, world 10
```

```
char *str = "hello, %s world\n";
printf(str);
```

- Format string need not be fixed.
- First argument is always format string.

```
$ ./a.out ↵
```

```
char *str = "hello, %s world\n";
printf(str, "\n");
```

- Format string need not be fixed.
- First argument is always format string.

```
$ ./a.out ↵
```

```
char *str = "hello, %s world\n";
printf(str, "\n");
```

- Format string need not be fixed.
- First argument is always format string.

```
$ ./a.out ↵
hello,
world
```

```
char *str = "hello, %s world\n";
printf(str, str);
```

- Format string need not be fixed.
- First argument is always format string.

```
$ ./a.out ↵
```

```
char *str = "hello, %s world\n";
printf(str, str);
```

- Format string need not be fixed.
- First argument is always format string.

```
$ ./a.out ↵
hello, hello, %s world
world
```

getchar(), putchar()

getchar(), putchar()

```
#include <stdio.h>


int main(void)
{




    return 0;
}
```

getchar(), putchar()

```
#include <stdio.h>


int main(void)
{
    int c;



    return 0;
}
```

getchar(), putchar()

```
#include <stdio.h>


int main(void)
{
    int c;
            c = getchar()

    return 0;
}
```

getchar(), putchar()

```
#include <stdio.h>


int main(void)
{
    int c;
           (c = getchar()) != EOF

    return 0;
}
```

getchar(), putchar()

```
#include <stdio.h>


int main(void)
{
    int c;
    while ((c = getchar()) != EOF)


    return 0;
}
```

getchar(), putchar()

```
#include <stdio.h>


int main(void)
{
    int c;
    while ((c = getchar()) != EOF)
        putchar(tolower(c));
    return 0;
}
```

getchar(), putchar()

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int c;

    while ((c = getchar()) != EOF)
        putchar(tolower(c));

    return 0;
}
```

getchar(), putchar()

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int c;

    while ((c = getchar()) != EOF)
        putchar(tolower(c));

    return 0;
}
```

	Returns	**Type**

getchar(), putchar()

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int c;

    while ((c = getchar()) != EOF)
        putchar(tolower(c));

    return 0;
}
```

	Returns	Type
getchar()		

getchar(), putchar()

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int c;

    while ((c = getchar()) != EOF)
        putchar(tolower(c));

    return 0;
}
```

	Returns	**Type**
getchar()	Character read or EOF	

getchar(), putchar()

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int c;

    while ((c = getchar()) != EOF)
        putchar(tolower(c));

    return 0;
}
```

	Returns	Type
getchar()	Character read or EOF	int

getchar(), putchar()

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int c;

    while ((c = getchar()) != EOF)
        putchar(tolower(c));

    return 0;
}
```

	Returns	Type
getchar()	Character read or EOF	int
putchar()		

getchar(), putchar()

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int c;

    while ((c = getchar()) != EOF)
        putchar(tolower(c));

    return 0;
}
```

	Returns	Type
getchar()	Character read or EOF	int
putchar()	Character written or EOF	

getchar(), putchar()

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    int c;

    while ((c = getchar()) != EOF)
        putchar(tolower(c));

    return 0;
}
```

	Returns	**Type**
getchar()	Character read or EOF	int
putchar()	Character written or EOF	int