

scanf()

scanf()

Task

Read input lines containing dates.

31 Mar 2016

scanf()

Task

Read input lines containing dates.

31 Mar 2016

Solution

```
int day, year;  
char monthname[20];  
  
scanf ( "%d %s %d" , &day, monthname, &year );
```

scanf()

Task

Read input lines containing dates.

31 Mar 2016

Solution

```
int day, year;  
char monthname[20];  
scanf ("%d %s %d", &day, monthname, &year );
```

scanf()

Task

Read input lines containing dates.

31 Mar 2016

Solution

```
int day, year;  
char monthname[20];  
scanf("%d %s %d", &day, monthname, &year);
```

Observations

- Function call.
- 

scanf()

Task

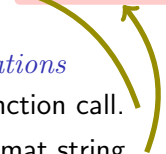
Read input lines containing dates.

31 Mar 2016

Solution

```
int day, year;  
char monthname[20];  
scanf("%d %s %d", &day, monthname, &year);
```

Observations

- Function call.
 - Format string.
- 

scanf()

Task


Read input lines containing dates.

31 Mar 2016

Solution

```
int day, year;  
char monthname[20];  
scanf("%d %s %d", &day, monthname, &year);
```

Observations

- Function call.
 - Format string.
 - Argument list.
- 

Task

Read input lines containing dates.

31/03/2016

Task

Read input lines containing dates.

31/03/2016

Solution

```
int day, month, year;
```

Task

Read input lines containing dates.

31/03/2016

Solution

```
int day, month, year;  
printf("Enter date :  ");
```

Task

Read input lines containing dates.

31/03/2016

Solution

```
int day, month, year;  
printf("Enter date : ");  
scanf (           , &day, &month, &year );
```

Task

Read input lines containing dates.

31/03/2016

Solution

```
int day, month, year;  
printf("Enter date : ");  
scanf ("%d/%s/%d", &day, &month, &year );
```

Task

Read input lines containing dates.

31/03/2016

Solution

```
int day, month, year;  
printf("Enter date : ");  
scanf ("%d / %s / %d", &day, &month, &year );
```

Task

Read input lines containing dates.

31/03/2016

Solution

```
int day, month, year;  
printf("Enter date : ");  
scanf ("%d / %s / %d", &day, &month, &year );  
printf("%d %d %d\n", day, month, year);
```

Task

Read input lines containing dates.

31/03/2016

Solution

```
int day, month, year;  
printf("Enter date : ");  
scanf ("%d / %s / %d", &day, &month, &year );  
printf("%d %d %d\n", day, month, year);
```

```
$ ./a.out ↵
```


Task

Read input lines containing dates.

31/03/2016

Solution

```
int day, month, year;  
printf("Enter date : ");  
scanf ("%d / %s / %d", &day, &month, &year );  
printf("%d %d %d\n", day, month, year);
```

```
$ ./a.out ↵
```

```
Enter date :
```

Task

Read input lines containing dates.

31/03/2016

Solution

```
int day, month, year;  
printf("Enter date : ");  
scanf ("%d / %s / %d", &day, &month, &year );  
printf("%d %d %d\n", day, month, year);
```

```
$ ./a.out ↵
```

```
Enter date : 31/03/2016 ↵
```

Task

Read input lines containing dates.

31/03/2016

Solution

```
int day, month, year;  
printf("Enter date : ");  
scanf ("%d / %s / %d", &day, &month, &year );  
printf("%d %d %d\n", day, month, year);
```

```
$ ./a.out ↵  
Enter date : 31/03/2016 ↵  
31 3 2016  
$
```

Task

Read input lines containing dates.

31/03/2016

Solution

```
int day, month, year;  
printf("Enter date : ");  
scanf ("%d / %s / %d", &day, &month, &year );  
printf("%d %d %d\n", day, month, year);
```

```
$ ./a.out ↵  
Enter date : 31/03/2016 ↵  
31 3 2016  
$
```

Question?

Why does this work?

```
scanf("%d_%d", &a, &b);
```

```
scanf("%d_%d", &a, &b);
```

```
scanf("%d__%d", &a, &b);
```

```
scanf("%d_%d", &a, &b);
```

```
scanf("%d__%d", &a, &b);
```

```
scanf("%d___%d", &a, &b);
```

```
scanf("%d_%d", &a, &b);
```

```
scanf("%d__%d", &a, &b);
```

```
scanf("%d___%d", &a, &b);
```

```
scanf("%d%d", &a, &b);
```


`scanf("%d_%d", &a, &b);` ✓

`scanf("%d__%d", &a, &b);` ✓

`scanf("%d___%d", &a, &b);` ✓

`scanf("%d%d", &a, &b);` ✓

`scanf("%d_%d", &a, &b);` ✓

`scanf("%d__%d", &a, &b);` ✓

`scanf("%d___%d", &a, &b);` ✓

`scanf("%d%d", &a, &b);` ✓

Rule

White spaces in format string is ignored.

```
scanf("%d__%d%d", &day, &month, &year);
```

```
i = scanf("%d__%d%d", &day, &month, &year);
```

```
i = scanf("%d__%d%d", &day, &month, &year);  
printf("i = %d\n", i);
```

```
i = scanf("%d_%d%d", &day, &month, &year);  
printf("i = %d\n", i);
```

```
$ ./a.out ↵
```

```
i = scanf("%d_%d%d", &day, &month, &year);  
printf("i = %d\n", i);
```

```
$ ./a.out ↵
```

```
2 ↵
```

```
i = scanf("%d_%d%d", &day, &month, &year);  
printf("i = %d\n", i);
```

```
$ ./a.out ↵
```

```
2 ↵
```

```
asdf ↵
```



```
i = scanf("%d_%d%d", &day, &month, &year);  
printf("i = %d\n", i);
```

```
$ ./a.out ↵  
2 ↵  
asdf ↵  
i = 1  
$
```

```
i = scanf("%d_%d%d", &day, &month, &year);  
printf("i = %d\n", i);
```

```
$ ./a.out ↵  
2 ↵  
asdf ↵  
i = 1  
$ ./a.out ↵
```

```
i = scanf("%d_%d%d", &day, &month, &year);  
printf("i = %d\n", i);
```

```
$ ./a.out ↵
```

```
2 ↵
```

```
asdf ↵
```

```
i = 1
```

```
$ ./a.out ↵
```

```
xyz ↵
```

```
i = scanf("%d_%d%d", &day, &month, &year);  
printf("i = %d\n", i);
```

```
$ ./a.out ↵  
2 ↵  
asdf ↵  
i = 1  
$ ./a.out ↵  
xyz ↵  
i = 0  
$
```

```
i = scanf("%d_%d%d", &day, &month, &year);  
printf("i = %d\n", i);
```

```
$ ./a.out ↵  
2 ↵  
asdf ↵  
i = 1  
$ ./a.out ↵  
xyz ↵  
i = 0  
$
```

Rule

`scanf()` returns the number of input correctly read.

```
printf("Enter name : ");  
scanf("%s", name);  
printf("Your name is %s\n.", name);
```

```
printf("Enter name : ");  
scanf("%s", name);  
printf("Your name is %s\n.", name);
```

```
$ ./a.out ↵
```

```
printf("Enter name : ");  
scanf("%s", name);  
printf("Your name is %s\n.", name);
```

```
$ ./a.out ↵  
Deepanjan_Kesh. ↵
```



```
printf("Enter name : ");  
scanf("%s", name);  
printf("Your name is %s\n.", name);
```

```
$ ./a.out ↵  
Deepanjan_Kesh. ↵  
Your name is
```

```
printf("Enter name : ");  
scanf("%s", name);  
printf("Your name is %s\n.", name);
```

```
$ ./a.out ↵  
Deepanjan_Kesh. ↵  
Your name is Deepanjan.  
$
```

```
printf("Enter name : ");  
scanf("%s", name);  
printf("Your name is %s\n.", name);
```

```
$ ./a.out ↵  
Deepanjan_Kesh. ↵  
Your name is Deepanjan.  
$
```

Rule

It reads upto white space.

```
printf("Enter name : ");  
scanf("%s", name);  
printf("Your name is %s\n.", name);
```

```
printf("Enter name : ");  
scanf("%[^\\n]s", name);  
printf("Your name is %s\\n.", name);
```

```
printf("Enter name : ");  
scanf("%[^\n]s", name);  
printf("Your name is %s\n.", name);
```

```
$ ./a.out ↵
```

```
printf("Enter name : ");  
scanf("%[^\n]s", name);  
printf("Your name is %s\n.", name);
```

```
$ ./a.out ↵  
Deepanjan_Kesh ↵
```

```
printf("Enter name : ");  
scanf("%[^\n]s", name);  
printf("Your name is %s\n.", name);
```

```
$ ./a.out ↵  
Deepanjan_Kesh ↵  
Your name is Deepanjan Kesh.  
$
```



```
printf("Enter name : ");  
scanf("%[^\\n]s", name);  
printf("Your name is %s\\n.", name);
```

```
$ ./a.out ↵  
Deepanjan_Kesh ↵  
Your name is Deepanjan Kesh.  
$
```

Rule

Read till you have not encountered '\\n'.

```
printf("Enter name : ");  
scanf("%[^\n]s", name);  
printf("Your name is %s\n.", name);
```

```
printf("Enter name : ");  
scanf("%[^p]s", name);  
printf("Your name is %s\n.", name);
```

```
printf("Enter name : ");  
scanf("%[^p]s", name);  
printf("Your name is %s\n.", name);
```

Rule

Read till you have not encountered 'p'.

```
printf("Enter name : ");  
scanf("%[^p]s", name);  
printf("Your name is %s\n.", name);
```

Rule

Read till you have not encountered 'p'.

$^p \rightarrow$ not 'p'.

```
printf("Enter name : ");  
scanf("%[^p]s", name);  
printf("Your name is %s\n.", name);
```

```
$ ./a.out ↵
```

Rule

Read till you have not encountered 'p'.

$\wedge p \rightarrow$ not 'p'.

```
printf("Enter name : ");  
scanf("%[^p]s", name);  
printf("Your name is %s\n.", name);
```

```
$ ./a.out ↵  
Deepanjan_Kesh ↵
```

Rule

Read till you have not encountered 'p'.

$^p \rightarrow$ not 'p'.

```
printf("Enter name : ");  
scanf("%[^p]s", name);  
printf("Your name is %s\n.", name);
```

```
$ ./a.out ↵  
Deepanjan_Kesh ↵  
Your name is Dee.  
$
```

Rule

Read till you have not encountered 'p'.

$^p \rightarrow$ not 'p'.


```
int a; int b;
```

```
int a; int b;  
scanf("%d", &a);  
scanf("%d", &b);
```

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

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

\$

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

```
$ ./a.out ↵
```

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

```
$ ./a.out ↵
```

```
1 2 ↵
```

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

```
$ ./a.out ↵  
1 2 ↵  
1 2  
$
```

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

```
$ ./a.out ↵  
1 2 ↵  
1 2  
$ ./a.out ↵
```



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

```
$ ./a.out ↵
```

```
1 2 ↵
```

```
1 2
```

```
$ ./a.out ↵
```

```
1
```

```
2
```

```
↵
```

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

```
$ ./a.out ↵
```

```
1 2 ↵
```

```
1 2
```

```
$ ./a.out ↵
```

```
1 2 ↵
```

```
1 2
```

```
$
```

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

```
$ ./a.out ↵
```

```
1 2 ↵
```

```
1 2
```

```
$ ./a.out ↵
```

```
1 2 ↵
```

```
1 2
```

```
$
```

Rules

- White spaces in input is also ignored.

```
int a; char b; int c;
```

```
int a; char b; int c;  
scanf("%d", &a);
```

```
int a; char b; int c;
```

```
scanf("%d", &a);
```

```
scanf("%c", &b);
```

```
int a; char b; int c;
```

```
scanf("%d", &a);
```

```
scanf("%c", &b);
```

```
scanf("%d", &c);
```

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```



```
int a; char b; int c;  
scanf("%d", &a);  
scanf("%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

\$

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵
```

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵
```

```
1 2 3 ↵
```

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵  
1 2 3 ↵  
1 2 ↵  
$
```

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵  
1 2 3 ↵  
1 2 ↵  
$
```

Questions

Why white space is not ignored?

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵  
1 2 3 ↵  
1 2 ↵  
$
```

Questions

Why white space is not ignored?

→ It is switched off for 'char'.

Solution

"`\%c`"

Solution

"_%c"

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

Solution

"_%c"

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

Solution

"_%c"

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("_%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

Solution

```
"_%c"
```

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("_%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

\$

Solution

"_%c"

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("_%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

\$./a.out ↵

Solution

```
"_%c"
```

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("_%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵
```

```
1 a 3 ↵
```

Solution

"_%c"

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("_%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵  
1 a 3 ↵  
1 a 3 ↵  
$
```


Solution

"_%c"

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("_%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵
```

```
1 a 3 ↵
```

```
1 a 3 ↵
```

```
$ ./a.out ↵
```

Solution

```
"_%c"
```

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("_%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵
```

```
1 a 3 ↵
```

```
1 a 3 ↵
```

```
$ ./a.out ↵
```

```
1 ↵
```

Solution

"_%c"

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("_%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵
```

```
1 a 3 ↵
```

```
1 a 3 ↵
```

```
$ ./a.out ↵
```

```
1 ↵
```

```
2 ↵
```

Solution

"_%c"

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("_%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵
```

```
1 a 3 ↵
```

```
1 a 3 ↵
```

```
$ ./a.out ↵
```

```
1 ↵
```

```
2 ↵
```

```
3 ↵
```

Solution

"_%c"

```
int a; char b; int c;  
scanf("%d", &a);  
scanf("_%c", &b);  
scanf("%d", &c);  
printf("%d %c %d\n", a, b, c);
```

```
$ ./a.out ↵
```

```
1 a 3 ↵
```

```
1 a 3 ↵
```

```
$ ./a.out ↵
```

```
1 ↵
```

```
2 ↵
```

```
3 ↵
```

```
1 2 3
```

White Spaces

White Spaces

Leading White space

White Spaces

Leading White space

```
%d      ignore    _ _23 ←      a = 23
```


White Spaces

Leading White space

%d	Ignore	__23 ↵	a = 23
%f	Ignore	___23.32 ↵	a = 23.32

White Spaces

Leading White space

%d	Ignore	__23 ↵	a = 23
%f	Ignore	___23.32 ↵	a = 23.32
%c	Do not ignore	___y ↵	a = _

White Spaces

Leading White space

%d	Ignore	__23 ↵	a = 23
%f	Ignore	___23.32 ↵	a = 23.32
%c	Do not ignore	___y ↵	a = _
%s	Ignore	___hello ↵	a = hello

White Spaces

White spaces in format strings

White Spaces

White spaces in format strings

Matches zero or more white spaces in input.

White Spaces

White spaces in format strings

Matches zero or more white spaces in input.

```
char c, str[20];  
scanf("%s%c", str, &c);  
printf("%s %c\n", str, c);
```

White Spaces

White spaces in format strings

Matches zero or more white spaces in input.

```
char c, str[20];  
scanf("%s%c", str, &c);  
printf("%s %c\n", str, c);
```

White Spaces

White spaces in format strings

Matches zero or more white spaces in input.

```
char c, str[20];  
scanf("%s%c", str, &c);  
printf("%s %c\n", str, c);
```

\$

White Spaces

White spaces in format strings

Matches zero or more white spaces in input.

```
char c, str[20];  
scanf("%s%c", str, &c);  
printf("%s %c\n", str, c);
```

```
$ ./a.out ↵
```

White Spaces

White spaces in format strings

Matches zero or more white spaces in input.

```
char c, str[20];  
scanf("%s%c", str, &c);  
printf("%s %c\n", str, c);
```

```
$ ./a.out ↵  
hello world ↵
```

White Spaces

White spaces in format strings

Matches zero or more white spaces in input.

```
char c, str[20];  
scanf("%s%c", str, &c);  
printf("%s %c\n", str, c);
```

```
$ ./a.out ↵  
hello world ↵  
hello ↵  
$
```

White Spaces

White spaces in format strings

Matches zero or more white spaces in input.

```
char c, str[20];  
scanf("%s %c", str, &c);  
printf("%s %c\n", str, c);
```

White Spaces

White spaces in format strings

Matches zero or more white spaces in input.

```
char c, str[20];  
scanf("%s %c", str, &c);  
printf("%s %c\n", str, c);
```

```
$ ./a.out ↵
```

White Spaces

White spaces in format strings

Matches zero or more white spaces in input.

```
char c, str[20];  
scanf("%s %c", str, &c);  
printf("%s %c\n", str, c);
```

```
$ ./a.out ↵  
hello world ↵
```

White Spaces

White spaces in format strings

Matches zero or more white spaces in input.

```
char c, str[20];  
scanf("%s %c", str, &c);  
printf("%s %c\n", str, c);
```

```
$ ./a.out ↵  
hello world ↵  
hello w  
$
```