scanf() in C

The scanf() function in C is the counterpart to printf() used to read input from the user. It can handle various data types such as integers, characters, strings, and other primitive types.

How to use scanf()?

The scanf() function reads input from the standard input device (keyboard) and stores it in variables.

Syntax

> scanf("format_string", address_of_arguments);

The **format_string** in scanf contains placeholders similar to printf, but it expects the addresses of variables as arguments. The address of any variable can be determined by using & addressof operator before the variable name.

> &variable_name

Example

C

```c
#include <stdio.h>
int main()
{
    int x, y;

    printf("Enter First Number: \n");
    scanf("%d", &x);

    printf("Enter Second Number: \n");
    scanf("%d", &y);

    printf("Sum is %d", x + y);
    return 0;
}
```

Output

```
Enter First Number:
10 (entered by user)
Enter Second Number:
20 (entered by user)
Sum is 30
```

Explanation:

When the above code is executed, it will print "Enter First Number: " and prints newline. When the program reaches a `scanf` statement, it pauses execution until the user provides input and presses the Enter key. For integer, it uses %d placeholder and address is determined using & operator.

Assume user enters 10 and press enter. Then it moves to the second printf line, prints "Enter Second Number:" and prints newline. Again, it waits for the user to enter the number. Assume user enter 20, then it prints the third printf statement "Sum is 30"

Strings and scanf()

Strings are stored as character arrays in C, so when reading strings, the variable name itself acts as the address (since strings are character arrays), so the & operator is not required.

C

```c
#include <stdio.h>
int main()
{
    char name[100];
    printf("Enter Your Name: \n");
    scanf("%s", name);

    printf("Hi %s,\n", name);
    printf("Wecome to GfG");
    return 0;
}
```

Output

```
Enter Your Name:
Sandeep (entered by user)
Hi Sandeep,
Welcome to GfG
```

However, scanf stops reading a string when it encounters a whitespace. For reading strings with spaces, use fgets instead.

C

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

```c
int main()
{
    char name[100];
    printf("Enter Your Name: \n");
    scanf("%s", name);

    printf("Hi %s,\n", name);
    printf("Wecome to GfG");
    return 0;
}
```

Output

```
Enter Your Name:
Sandeep Jain(entered by user)
Hi Sandeep,
Welcome to GfG
```

Multiple Inputs with scanf

You can read multiple values in a single scanf statement by specifying multiple placeholders.

C

```c
#include <stdio.h>
int main()
{
    int x, y;

    printf("Enter Two Number: \n");
    scanf("%d %d", &x, &y);

    printf("Multiplication is %d", x * y);
    return 0;
}
```

Output

```
Enter Two Number:
10 20 (entered by user)
Multiplication is 200
```

The user can provide inputs either by separating values with spaces or pressing Enter between inputs.

Input Format with Separators

If you include separators (e.g., strings) in the format string, the user must input the exact separator for the scanf function to work correctly.

C

```c
#include <stdio.h>
int main()
{
    int x, y;

    printf("Enter Two Number: \n");
    scanf("%dGfG%d", &x, &y);

    printf("Multiplication is %d", x * y);
    return 0;
}
```

Output

```
Enter Two Number:
10GfG20 (entered by user)
Multiplication is 200
```

If the input doesn't match the format (e.g., missing **GfG**), the program may produce incorrect results or errors.