

Difference Between Printf and Scanf:

Scanf is used to take in input from the console. Printf is used to print output to the console. Console here refers to the input and output windows you see on prutor for the purpose of solving questions on Prutor. You will learn about what it actually means later.

Format specifiers

The string snippets seen in the print and scan statements are called **Format Specifiers**. Some examples are: `%d`, `%c`, `%f`. They are used to specify to the `printf()` or `scanf()` how to interpret the type of the variable to be printed or scanned.

`%d` → used for `int` variables (integers)

`%c` → used for `char` variables (characters)

`%f` → used for `float` variables (decimal point numbers)

Note: For `printf()`, there is no difference between using `%d` and `%i` specifiers. The difference is only in `scanf()` where `%i` detects the base of input as well. (This is an advanced concept. `%d` suffices for most use cases.)

Scanning multiple inputs in the same line

You can use `scanf()` for scanning multiple inputs. The line,

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

takes input an integer up till it encounters a non-digit character (eg. space or new line, or any other character) in the input. (Another advanced fact is that it ignores multiple spaces between the inputs.) Example, if the input is:

```
11 22 33
```

and we scan 3 times:

```
scanf("%d", &x);  
scanf("%d", &y);  
scanf("%d", &z);
```

Then each `scanf` will read an integer until it encounters a space, so it reads 11 and stores it in `x`, it reads 22 into `y`, and 33 into `z`. So, if you print them,

```
printf("x = %d, y = %d, z = %d", x, y, z);
```

The output is: x = 11, y = 22, z = 33

You can use this simple technique to take multiple inputs given in the same line.

Runtime Errors

A runtime error in a program is an error that occurs while the program is running after being successfully compiled. Some common examples of runtime errors that you may encounter:

1. Division by 0:

If your program does division by 0 at any time, though it will compile correctly (meaning that the code is correct), it cannot do the calculation while executing the program, giving a runtime error. Example code (will give Runtime error at $z = x/y$):

```
#include<stdio.h>

int main(){
    int x = 10;
    int y = 0;
    int z;
    z = x/y;
}
```

2. Forgetting & in scanf():

This is a common error. The reason for putting & will be clear later in the future. For now, just remember adding & before variable names. Example (following will give an error):

```
#include<stdio.h>

int main(){
    int x;
    scanf("%d", x);
}
```

The corrected statement will be `scanf("%d", &x);`

Difference between Float and Int Variables

1. An integer variable can only save integers, while floats can save floating-point (numbers with decimal points, i.e. 5.67) numbers as well
2. Integer variables when multiplied with float constant result in integers:
 - a. `int a = 5; a = a*4.1; // here a will be 20 and NOT 20.5`
3. Float variables when multiplied with float constants with result in floats:
 - a. `float a = 5; a = a*4.1; // here a will be 20.5`
4. An Integer when divided by an integer results in the quotient of the division
 - a. `int a = 5; a = a/2; // here a is 2 and not 2.5`
 - b. `int a = 5; float b = a/2; // here b is also 2 and not 2.5`
5. **DUE TO THESE REASONS PLEASE MAKE SURE THAT YOU ARE USING FLOAT VARIABLES WHEN DEALING WITH DECIMAL NUMBERS.**

Using Printf with Floats:

1. `float a = 5.66; printf("%f", a); //prints 5.660000 to console. Here by default 6 decimal places are printed.`
2. IF YOU WANT TO PRINT ONLY SOME DECIMAL PLACES, USE: `printf("%.3f", a); // prints 5.660`
3. **NOTE that using the above way, will also round off, i.e, `printf("%.1f", a); // prints 5.7`**

Difference between `int main()` and `void main()`

A	B
<pre>int main(){ ... return 0; }</pre>	<pre>void main(){ }</pre>

Both the syntax for the `main()` method are valid. The `void main()` indicates that the `main()` function will not return any value, but the `int main()` indicates that the `main()` can return integer type data.

NOTE: It is good practice to use `int main()` for developing C programs. **The Prutor IDE reports a warning for syntax B. It is advised to use syntax A.**