**1.Go to the printf implementation and find out why we are getting junk ?**

Printf working principle

printf or print function in C takes a formatting string and couple of optional variables as input and outputs strings to console while converting input variables to strings.

Printf block diagram



Printf and scanf takes multiple arguments and these functions are called variable length arguments function or vararg function. Take printf for consideration. User supply a string and input argumnets. Printf creates an internal buffer for constructing output string. Now printf iterates through each characters of user string and copies the character to the output string. Printf only stops at "%". "%" means there is an argument to convert. Arguments are in the form of char, int, long, float, double or string. It converts it to string and appends to output buffer. If the argument is string then it does a string copy. Finally printf may reach at the end of user sting and it copies the entire buffer to the stdout file.

Implement printf

Let us implement our own printf function. This is only for the understanding purpose. We name it print(). It has one string argument (str) and rest are variable arguments. Variable arguments are managed by macros like va\_start, va\_arg and va\_end. A temporary buffer (buff) is there to construct the output buffer. A while loop is needed to scan each characters in the input string. Now we iterate character by character in the loop and copy each character to output string. Same time we check for "%". "%" is not copied to output string. Once we found it, we check the next character. This is the formatting character. Formatting character says how to format the argument to visible output string. Printf supports varieties of formatting. C is for character, d for decimal integer, f for floating point, x for hexadecimal and s for strings. We match the formatting and pick the argument variable using va\_arg(). Argument variable is then converted to string format and appends to the output string. character can be copied as it is and Itoa function is used for integer to string conversion.

**2.Conclude structure copy of C is shallow copy or deep copy?**

Shallow Copy −

A shallow copy of an object copies the "main" object, but doesn’t copy the inner objects.The "inner objects" are shared between the original object and its copy.The problem with the shallow copy is that the two objects are not independent. If you modify the one object, the change will be reflected in the other object.

Deep Copy −

A deep copy is a fully independent copy of an object. If we copied our object, we would copy the entire object structure.

**3.Draw the memorry layout for big & little endian and map the variables from the given program!**

Endianness refers to the bytes order in which data stored in the memory and also describes the order of byte transmission over a digital link. Basically Endianness comes in two varieties little endian and big endian and in which order data will be stored in memory it depends on the endianness.

If your machine is big-endian then the MSB byte store first (means at lower address) and if the machine is the little-endian then LSB byte store first (means at lower address).



**4.List out the applications of Union?**

The primary use of a union is allowing access to a common location by different data types, for example hardware input/output access, bitfield and word sharing, or type punning. Unions can also provide low-level polymorphism.