# Function Parameter Passing and Return Assignments

**Mandatory**

1. Refer the code below and find the issue.

#include<stdio.h>

int \*func(void);

int main()

{

int num,\*ptr = NULL;

ptr = (int \*)func();

num = \*ptr;

return 1;

}

int \*func()

{

int local;

local = 10;

return(&local);

}

In above code is there a way(s) to return local variable address to caller?

**Ans**: The issue with the given code lies in the way the local variable local is handled in the function func(). Specifically, the code attempts to return the address of the local variable local to the caller, which is problematic because local is a **local variable** and its memory is deallocated when the function func() returns. This causes undefined behaviour when attempting to dereference the pointer returned by func() in main().

To avoid returning the address of a local variable, we can do one of the following:

1. Use a static Variable
2. Dynamically Allocate Memory
3. Pass a pointer to the function

A screenshot of a computer

Description automatically generated

1. Write a program with a function ***read\_extract\_characters()*** to read a string (of max length 50 characters) from user, extract the characters at odd indices, store in an other array and return to the call. Caller should be able to read and display the extracted string.

[Note : do not return a local variable in function to caller]

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Write below functions to extract and return the required inputs from an email id string of max length 80 characters. Program should be able to detect an invalid email id too based on following validations. Also value returned should be in scope and accessible in caller.
   1. valid email id will have username, host and domain name (as .com/.edu)

A screenshot of a computer

Description automatically generated

