**Function Parameter Passing and Return Assignments**

**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?**

The issue with the code is that it attempts to return the address of a local variable from the function func():

You cannot return the address of a local variable from a function. Instead, you have a few options:

1. Return a dynamically allocated memory address (heap memory).
2. Pass a pointer to the function and have the function modify it.

#include<stdio.h>

#include<stdlib.h> // Include for malloc

int \*func(void);

int main()

{

int num, \*ptr = NULL;

ptr = func(); // Now func returns a dynamically allocated address

num = \*ptr; // Dereference the valid address

printf("num = %d\n", num);

// Don't forget to free the memory

free(ptr);

return 0;

}

int \*func()

{

int \*local = malloc(sizeof(int)); // Dynamically allocate memory

if (local == NULL) {

printf("Memory allocation failed\n");

return NULL; // Return NULL in case of memory allocation failure

}

\*local = 10; // Assign value to dynamically allocated memory

return local; // Return pointer to dynamically allocated memory

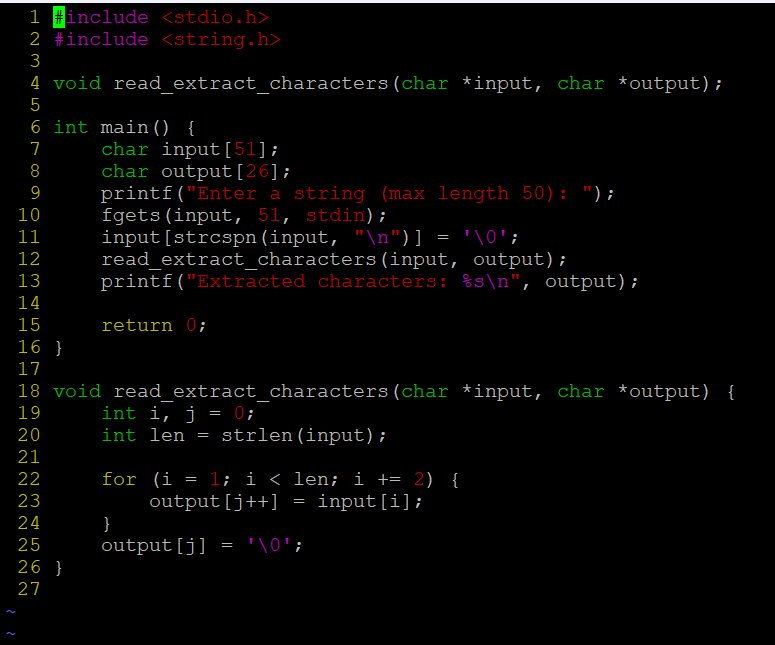
}

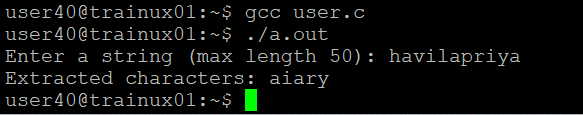
In this version, we allocate memory dynamically using malloc and return the pointer to the caller. The caller is then responsible for freeing the memory when it is no longer needed.

**2. 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],**

**ANS:**





**3. 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.**

a. valid email id will have username, host and domain name (as .com/.edu)

Functions:

char \*getuser(char input[]); // return NULL or valid username from email id input

char \*gethost(char input[]); //return NULL or valid hostname from email id input

char \*getdomain(char input[]); //return NULL or valid domain name from email id input

int isValidDomain(char input[]); // return 1 if domain is “.com” or “.edu” else 0

A screen shot of a computer program

Description automatically generated

