**Function Parameter Passing and Return Assignments**

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?

A:

1. Using a **static variable**, which persists after the function ends.

#include<stdio.h>

int \*func(void);

int main()

{

int num, \*ptr = NULL;

ptr = func(); // No need for typecast, it's already an int\*

num = \*ptr;

printf("%d\n", num); // To verify the result

return 0;

}

int \*func()

{

static int local; // Make the variable static

local = 10;

return &local; // Valid, because 'local' is static

}

1. Using **dynamic memory allocation** (e.g., malloc) to ensure the memory persists after the function returns.

#include<stdio.h>

#include<stdlib.h>

int \*func(void);

int main()

{

int num, \*ptr = NULL;

ptr = func(); // No need for typecast

num = \*ptr;

printf("%d\n", num); // To verify the result

free(ptr); // Don't forget to free the dynamically allocated memory

return 0;

}

int \*func()

{

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

if (local != NULL) {

\*local = 10;

}

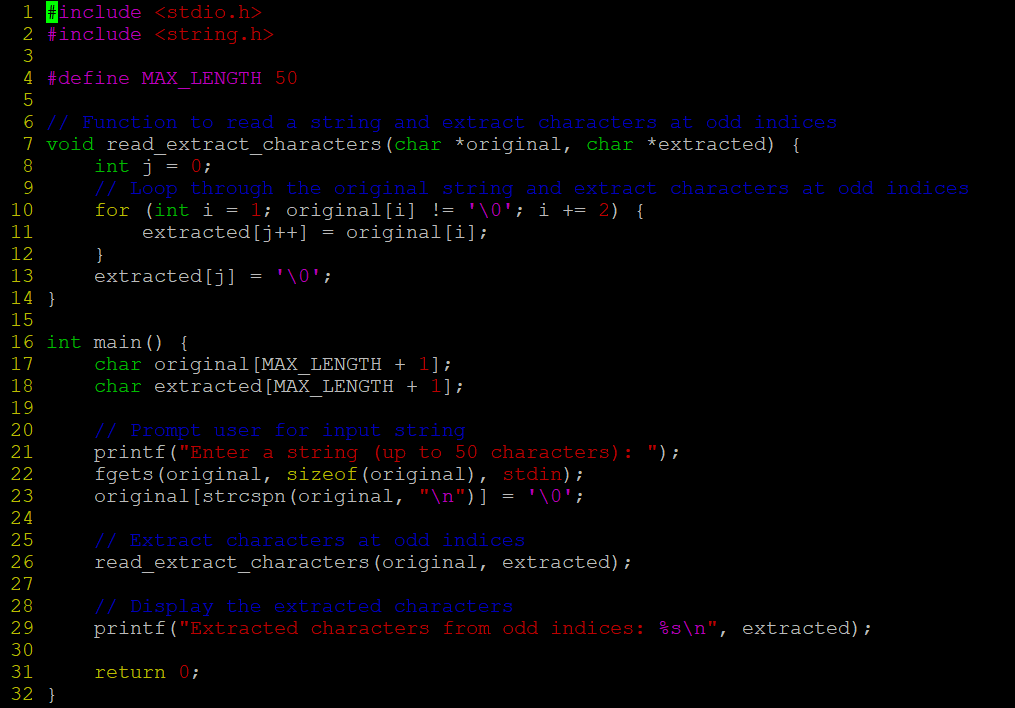
return local; // Valid address

}

Both approaches allow you to safely return a pointer from the function and use it in the caller.

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]



OUTPUT:

A black background with white text

Description automatically generated

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.

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

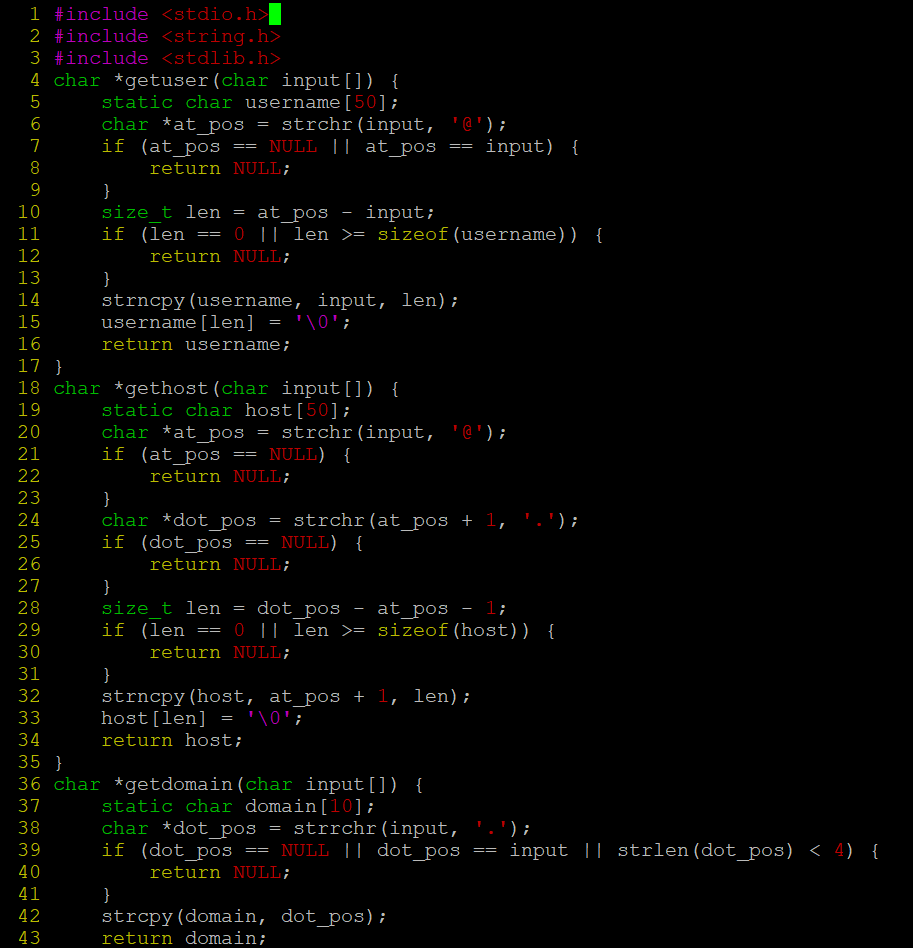
**Functions**:

*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*

**

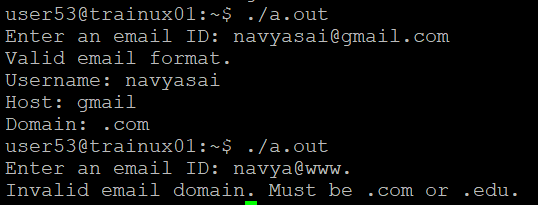
*A screen shot of a computer program

Description automatically generated*

*A black screen with red text

Description automatically generated*

*OUTPUT:*

**