**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?

1. The issue in the code is that returning the address of a local variable local is unsafe because it is allocated on the stack, and its memory is deallocated when the function scope ends.

**Solution 1: Use static keyword**

#include<stdio.h>

int \*func(void);

int main()

{

int num, \*ptr = NULL;

ptr = func();

num = \*ptr;

printf("Value: %d\n", num);

return 0;

}

int \*func()

{

static int local; // Use static to persist variable after function returns

local = 10;

return &local;

}

**Solution 2: Use dynamic memory allocation (malloc)**

#include<stdio.h>

#include<stdlib.h>

int \*func(void);

int main()

{

int num, \*ptr = NULL;

ptr = func();

num = \*ptr;

printf("Value: %d\n", num);

free(ptr); // Free allocated memory

return 0;

}

int \*func()

{

int \*local = (int \*)malloc(sizeof(int));

\*local = 10;

return local;

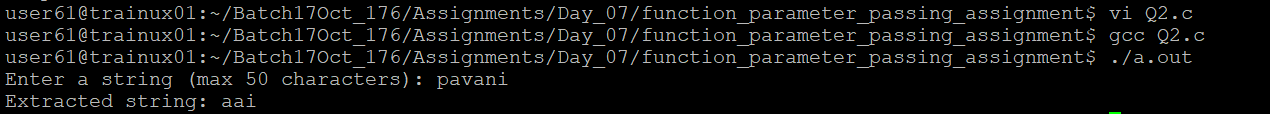
}

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]

A screenshot of a computer program

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.

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

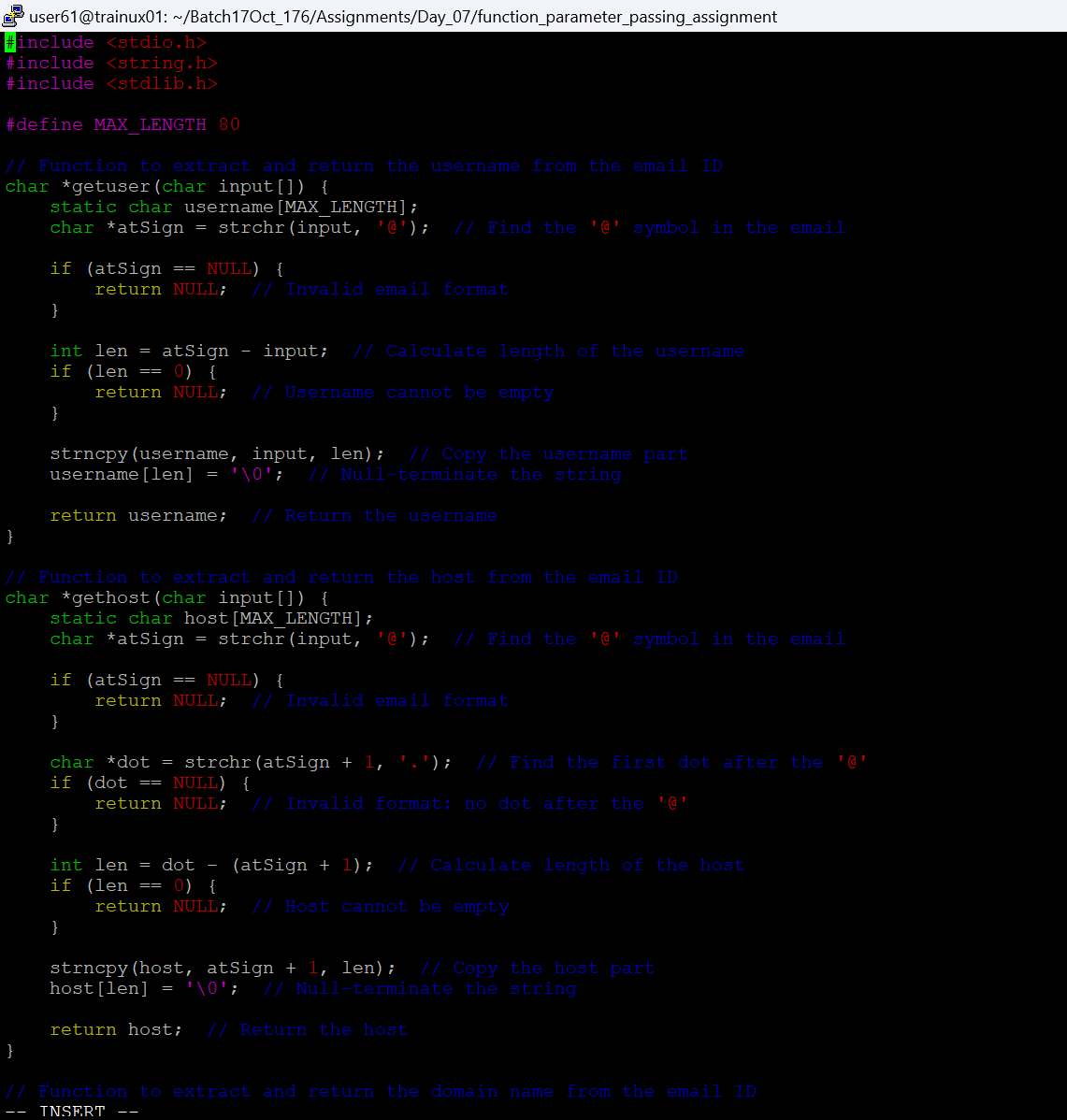
Functions:

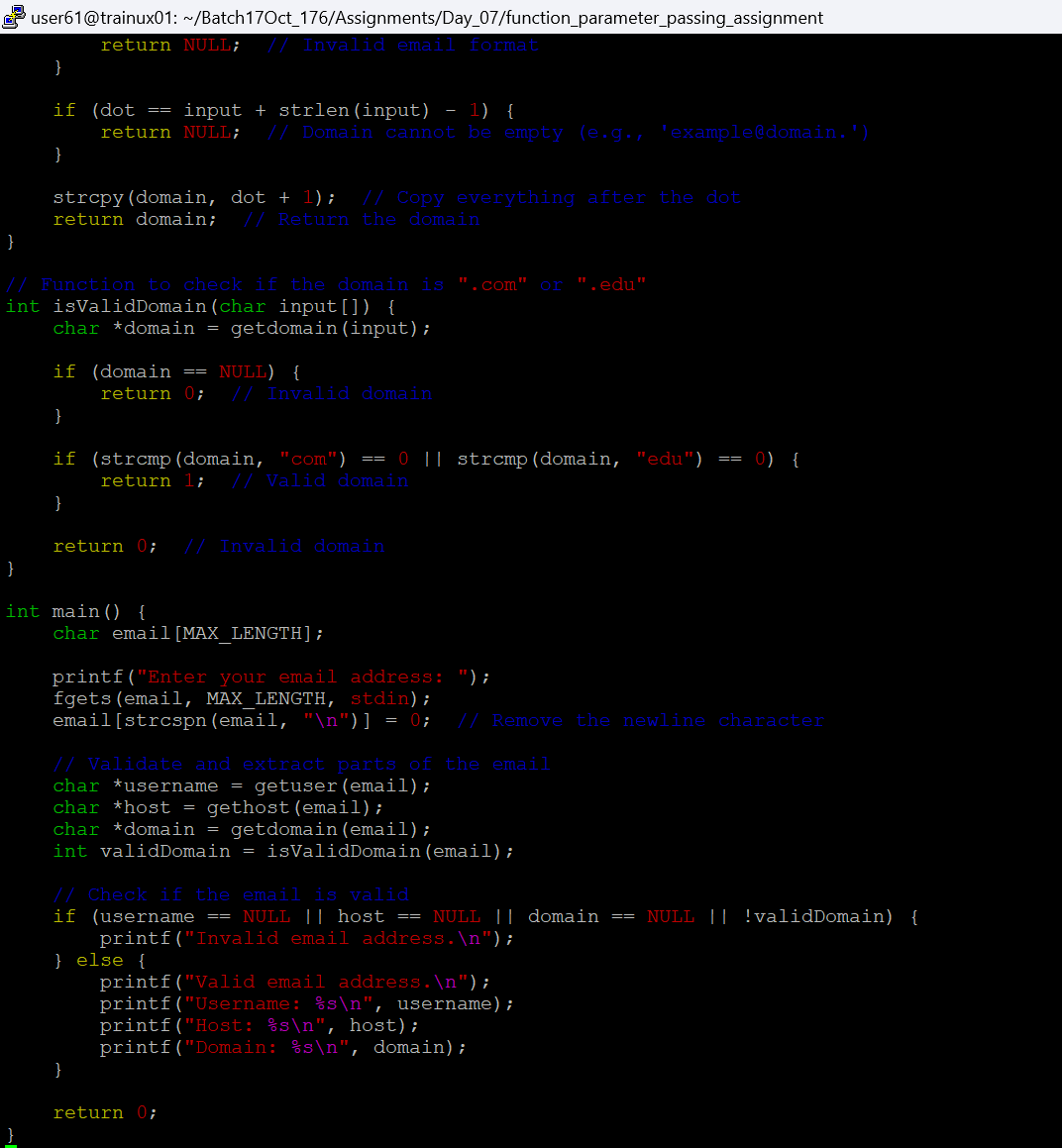
char \*getuser(char input[]); // return NULL or valid username from email id input

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 computer screen with white text

Description automatically generated