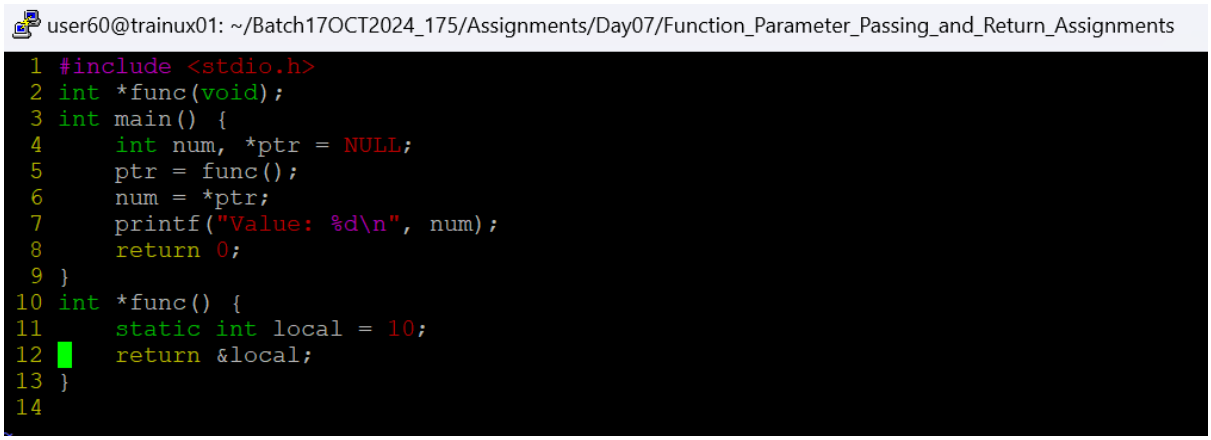
Function Parameter Passing and Return Assignments

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?



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]

user60@trainux01: ~/Batch17OCT2024_175/Assignments/Day07/Function_Parameter_Passing_and_Return_Assignments

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char* read_extract_characters() {
    char input[51];
    printf("Enter a string (max 50 characters): ");
    fgets(input, sizeof(input), stdin);
    input[strcspn(input, "\n")] = '\0';
    int length = strlen(input);
    int newLength = (length + 1) / 2;
    char *extracted = (char *)malloc(newLength + 1);

    int j = 0;
    for (int i = 1; i < length; i += 2) {
        extracted[j++] = input[i];
    }
    extracted[j] = '\0';

    return extracted;
}

int main() {
    char *oddChars = read_extract_characters();
    printf("Extracted characters at odd indices: %s\n", oddChars);
    free(oddChars);

    return 0;
```

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

user60@trainux01: ~/Batch17OCT2024_175/Assignments/Day07/Function_Parameter_Passing_and_Return_Assignments

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define MAX_EMAIL_LEN 80
char *getuser(char input[]) {
    static char username[MAX_EMAIL_LEN];
    char *atSign = strchr(input, '@');
    if (atSign != NULL) {
        size_t len = atSign - input;
        strncpy(username, input, len);
        username[len] = '\0';
        return username;
    }
    return NULL;
}
char *gethost(char input[]) {
    static char host[MAX_EMAIL_LEN];
    char *atSign = strchr(input, '@');
    if (atSign != NULL) {
        char *dot = strchr(atSign + 1, '.');
        if (dot != NULL) {
            size_t len = dot - atSign - 1;
            strncpy(host, atSign + 1, len);
            host[len] = '\0';
            return host;
        }
    }
    return NULL;
}
char *getdomain(char input[]) {
    static char domain[MAX_EMAIL_LEN];
    char *dot = strrchr(input, '.');
    if (dot != NULL && (strcmp(dot, ".com") == 0 || strcmp(dot, ".edu") == 0)) {
        strcpy(domain, dot);
        return domain;
    }
    return NULL;
}
int isValidDomain(char input[]) {
    char *domain = getdomain(input);
    return (domain != NULL);
}
int main() {
    char email[MAX_EMAIL_LEN];
    printf("Enter an email address (max 80 characters): ");
    fgets(email, sizeof(email), stdin);
    email[strcspn(email, "\n")] = '\0';
    if (isValidDomain(email)) {
        printf("Valid email!\n");

        char *username = getuser(email);
        char *host = gethost(email);
        char *domain = getdomain(email);

        printf("Username: %s\n", username);
        printf("Host: %s\n", host);
        printf("Domain: %s\n", domain);
    } else {
        printf("Invalid email address.\n");
    }
    return 0;
}
```

```
user72@trainux01:~/splint$ vi fun.c
user72@trainux01:~/splint$ gcc fun.c
user72@trainux01:~/splint$ ./a.out
Enter an email address (max 80 characters): chaithrakenchanna15@gmail.com
Valid email!
Username: chaithrakenchanna15
Host: gmail
Domain: .com
```