

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);
}
```

- A. In the above code there is a segmentation fault. As the local variable cannot be returned. In above code is there a way(s) to return local variable address to caller?

Ans:

By using the static keyword we can return local variable. As the variable stored throughout the process

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]

```

#include<stdio.h>
#include<string.h>
char *read_exact_characters(char *);
int main()
{
    char arr[50];
    printf("enter array");
    scanf("%s",arr);
    printf("array is:%s\n",read_exact_characters(arr));
}
char* read_exact_characters(char * arr)
{
    static char a[50];
    int i,j;
    for(i=1,j=0;i<strlen(arr);i=i+2,j++)
    {
        a[j]=arr[i];
    }
    return a;
}
~
~
~

```

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

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
char * getuser(char []);
char * gethost(char []);
char * getdomain(char []);
int isValidDomain(char []);
int main() {
    char email[81];

    printf("Enter email id: ");
    fgets(email, 81, stdin);
    email[strcspn(email, "\n")] = '\0';

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

    if (username != NULL && hostname != NULL && domain != NULL && isValidDomain(email)) {
        printf("Valid email!\n");
        printf("Username: %s\n", username);
        printf("Host: %s\n", hostname);
        printf("Domain: %s\n", domain);
    } else {
        printf("Invalid email format!\n");
    }

    return 0;
}

```

```

char* getuser(char input[]) {
    int i = 0;

    static char username[50];
    char *at_pos = strchr(input, '@');
    if (at_pos == NULL) {
        return NULL;
    }

    for (i = 0; i < (at_pos - input); i++) {
        username[i] = input[i];
    }
    username[i] = '\0';

    if (i == 0) {
        return NULL;
    }

    return username;
}

char* gethost(char input[]) {
    static char host[81];
    int i = 0;

    char *at_pos = strchr(input, '@');
    char *dot_pos = strchr(at_pos + 1, '.');

    if (at_pos == NULL || dot_pos == NULL) {
        return NULL;
    }

    for (i = 0; at_pos + 1 + i < dot_pos; i++) {
        host[i] = at_pos[1 + i];
    }
    host[i] = '\0';

    if (i == 0) {
        return NULL;
    }

    return host;
}

```

```

char* getdomain(char input[]) {
    static char domain[10];
    char *dot_pos = strchr(input, '.');

    if (dot_pos == NULL) {
        return NULL;
    }

    strcpy(domain, dot_pos + 1);

    if (strlen(domain) == 0) {
        return NULL;
    }

    return domain;
}

int isValidDomain(char input[]) {
    char *domain = getdomain(input);
    if (domain != NULL) {
        if (strcmp(domain, "com") == 0 || strcmp(domain, "edu") == 0) {
            return 1;
        }
    }
    return 0;
}

```