iEnumerated Datatypes and Typedefs assignments

Mandatory

1. WAP to define an enum to store designations in an organization. List of possible values are

   {E2F=1, E2, E3, E4, E5}

   Prompt and read a designation from the user. Then display his designation string such as

Designation	Designation String
E2F	Software Fresher
E2	Software Engineer
E3	Senior Software Engineer
E4	Team Lead
E5	Senior Team Lead

user72@trainux01: ~/Assignments

```c
#include <stdio.h>

typedef enum {
    E2F = 1, E2, E3, E4, E5
} Designation;
const char* getDesignationString(Designation designation) {
    switch(designation) {
        case E2F:
        return "Software Fresher";
        case E2:
        return "Software Engineer";
        case E3:
        return "Senior Software Engineer";
        case E4:
        return "Team Lead";
        case E5:
        return "Senior Team Lead";
        default:
        return "Unknown Designation";
    }
}
int main() {
    int input;
    printf("Enter the designation (1 for E2F, 2 for E2, 3 for E3, 4 for E4, 5 for E5): ");
    scanf("%d", &input);
    if (input < E2F || input > E5) {
        printf("Invalid input.\n");
        return 1;
    }
    Designation designation = (Designation)input;
    printf("Designation: %d\n", designation);
    printf("Designation String: %s\n", getDesignationString(designation));
    return 0;
}
```

```
user72@trainux01:~/Assignments$ vi proc1.c
user72@trainux01:~/Assignments$ gcc proc1.c
user72@trainux01:~/Assignments$ ./a.out
Enter the designation (1 for E2F, 2 for E2, 3 for E3, 4 for E4, 5 for E5): 3
Designation: 3
Designation String: Senior Software Engineer
```

2. Define a typedef structure to keep the configuration of putty server. Identify and place all the required members. Create a structure variable and initialize it with user defined values and finally display the contents.

user72@trainux01: ~/Assignments

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

typedef struct {
    char host[100];
    int port;
    char username[50];
    char password[50];
    int timeout;
    int ssh_enabled;
} PuttyServerConfig;

int main() {
    PuttyServerConfig config;
    strcpy(config.host, "192.168.1.100");
    config.port = 22;
    strcpy(config.username, "admin");
    strcpy(config.password, "password123");
    config.timeout = 300;
    config.ssh_enabled = 1;
    printf("Putty Server Configuration:\n");
    printf("Host: %s\n", config.host);
    printf("Port: %d\n", config.port);
    printf("Username: %s\n", config.username);
    printf("Password: %s\n", config.password);
    printf("Timeout: %d seconds\n", config.timeout);
    printf("SSH Enabled: %s\n", config.ssh_enabled ? "Yes" : "No");
    return 0;
}
```

```
user72@trainux01:~/Assignments$ vi proc2.c
user72@trainux01:~/Assignments$ gcc proc2.c
user72@trainux01:~/Assignments$ ./a.out
Putty Server Configuration:
Host: 192.168.1.100
Port: 22
Username: admin
Password: password123
Timeout: 300 seconds
SSH Enabled: Yes
```