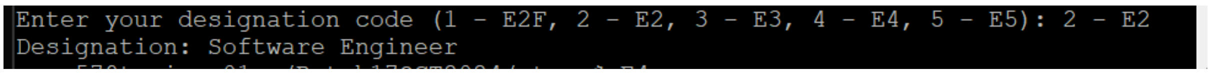
Enumerated Datatypes and Typedefs assignments

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

Sol:

```
#include <stdio.h>

typedef enum {

   E2F = 1,      // 1: Software Fresher

   E2,          // 2: Software Engineer

   E3,          // 3: Senior Software Engineer

   E4,          // 4: Team Lead

   E5          // 5: Senior Team Lead

} Designation;


const char* getDesignationString(Designation des) {

   switch (des) {

      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 "Invalid Designation";
    }
}


int main() {
    int inputDesignation;


    printf("Enter your designation code (1 - E2F, 2 - E2, 3 - E3, 4 - E4, 5 - E5): ");
    scanf("%d", &inputDesignation);


     if (inputDesignation < 1 || inputDesignation > 5) {
        printf("Invalid designation code entered!\n");
     } else {


      Designation des = (Designation)inputDesignation;
      printf("Designation: %s\n", getDesignationString(des));
    }


    return 0;
}
```

Output:



```
Enter your designation code (1 - E2F, 2 - E2, 3 - E3, 4 - E4, 5 - E5): 2 - E2
Designation: Software Engineer
```

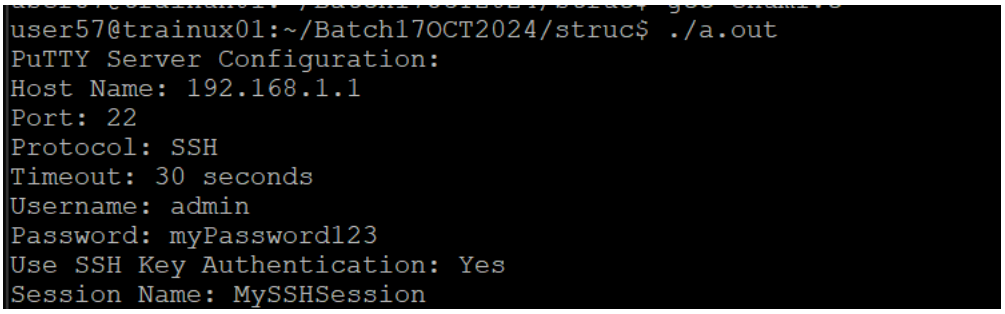
2. Define a typedef structure to keep the configuration of putty server. Identify and place all the required members. Create a structure variable and initialize it with user defined values and finally display the contents.

Sol:

```
#include <stdio.h>

#include <string.h>
```

```
typedef struct {
    char hostName[100];
    int port;
    char protocol[10];
    int timeout;
    char username[50];
    char password[50];
    int useSSHKey;
    char sessionName[100];
} PuttyConfig;



void displayConfig(PuttyConfig config) {
    printf("PuTTY Server Configuration:\n");
    printf("Host Name: %s\n", config.hostName);
    printf("Port: %d\n", config.port);
    printf("Protocol: %s\n", config.protocol);
    printf("Timeout: %d seconds\n", config.timeout);
    printf("Username: %s\n", config.username);
    printf("Password: %s\n", config.password);
    printf("Use SSH Key Authentication: %s\n", config.useSSHKey ? "Yes" : "No");
    printf("Session Name: %s\n", config.sessionName);
}

int main() {
    PuttyConfig myConfig = {
        "192.168.1.1",   // Host name
        22,           // Port (22 is the default SSH port)
        "SSH",         // Protocol
        30,           // Timeout (in seconds)
        "admin",        // Username
        "myPassword123", // Password (in real use, this should be handled securely)
        1,              // Use SSH Key (1 means Yes)
        "MySSHSession"   // Session Name
    };
    displayConfig(myConfig);


    return 0;
}
```

Output:



```
user57@trainux01:~/Batch17OCT2024/struc$ ./a.out
PuTTY Server Configuration:
Host Name: 192.168.1.1
Port: 22
Protocol: SSH
Timeout: 30 seconds
Username: admin
Password: myPassword123
Use SSH Key Authentication: Yes
Session Name: MySSHSession
```