

addnum.x

```
struct addnum
{
    int a;
    int b;
};

program ADDNUM_PROG
{
    version ADDNUM_VERS
    {
        int ADDNUM(addnum) = 1;

    } = 1;

} = 0x11111111;
```

addnum_server.c

```
/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "addnum.h"

int *
addnum_1_svc(addnum *argp, struct svc_req *rqstp)
{
    static int result,n1,n2,sum;//Line no. 12

    n1=argp->a;
    n2=argp->b;

    sum=n1+n2;

    result=sum;

    return &result;
}
```

addnum_client.c

```
/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "addnum.h"
```

```

void
addnum_prog_1(char *host,int a,int b)//Line no. 11
{
    CLIENT *clnt;
    int *result_1;
    addnum addnum_1_arg;

#ifdef DEBUG
    clnt = clnt_create (host, ADDNUM_PROG, ADDNUM_VERS, "udp");
    if (clnt == NULL) {
        clnt_pcreateerror (host);
        exit (1);
    }
#endif /* DEBUG */
    addnum_1_arg.a=a;//Line no. 24
    addnum_1_arg.b=b;
    result_1 = addnum_1(&addnum_1_arg, clnt);
    if (result_1 == (int *) NULL) {
        clnt_perror (clnt, "call failed");
    }
    printf("\n Server returns=%d", *result_1);//Line no. 29
#ifdef DEBUG
    clnt_destroy (clnt);
#endif /* DEBUG */
}

```

```

int
main (int argc, char *argv[])
{
    char *host;
    int a,b,ch; //Line no. 39
    if (argc < 2) {
        printf ("usage: %s server_host\n", argv[0]);
        exit (1);
    }
    host = argv[1];
do//Line no. 45
{
    system("clear");
    printf("\nEnter a no. 1: ");
    scanf("%d",&a);
    printf("\nEnter a no. 2: ");
    scanf("%d",&b);
    addnum_prog_1(host,a,b);

    printf("\nTry again : (1/0) :: ");
    scanf("%d",&ch);

}while(ch==1);
exit (0);
}

```

```
(base) student@student-ASUS-EXPERTCENTER-D500MD-D500MD-IN:~/Documents/r$  
./addnum_server
```

```
(base) student@student-ASUS-EXPERTCENTER-D500MD-D500MD-IN:~/Documents/r$  
./addnum_client localhost
```

Enter a no. 1: 2

Enter a no. 2: 3

Server returns=5

Try again : (1/0) :: 0