Code:

x-file:

```
struct nums {
   int x;
   int y;
};

program ADD_PROG {
   version ADD_VERS {
      int add(nums)=1;
   }=1;
}=0x12341234;
```

Client:

```
/*
 * Please do not edit this file.
 * It was generated using rpcgen.
 */

#include <memory.h> /* for memset */
#include "add.h"

/* Default timeout can be changed using clnt_control() */
static struct timeval TIMEOUT = { 25, 0 };

int *
add_1(nums *argp, CLIENT *clnt)
{
	static int clnt_res;

	memset((char *)&clnt_res, 0, sizeof(clnt_res));
	if (clnt_call (clnt, add,
		(xdrproc_t) xdr_nums, (caddr_t) argp,
		(xdrproc_t) xdr_int, (caddr_t) &clnt_res,
		TIMEOUT) != RPC_SUCCESS) {
		return (NULL);
	}
	return (&clnt_res);
}
```

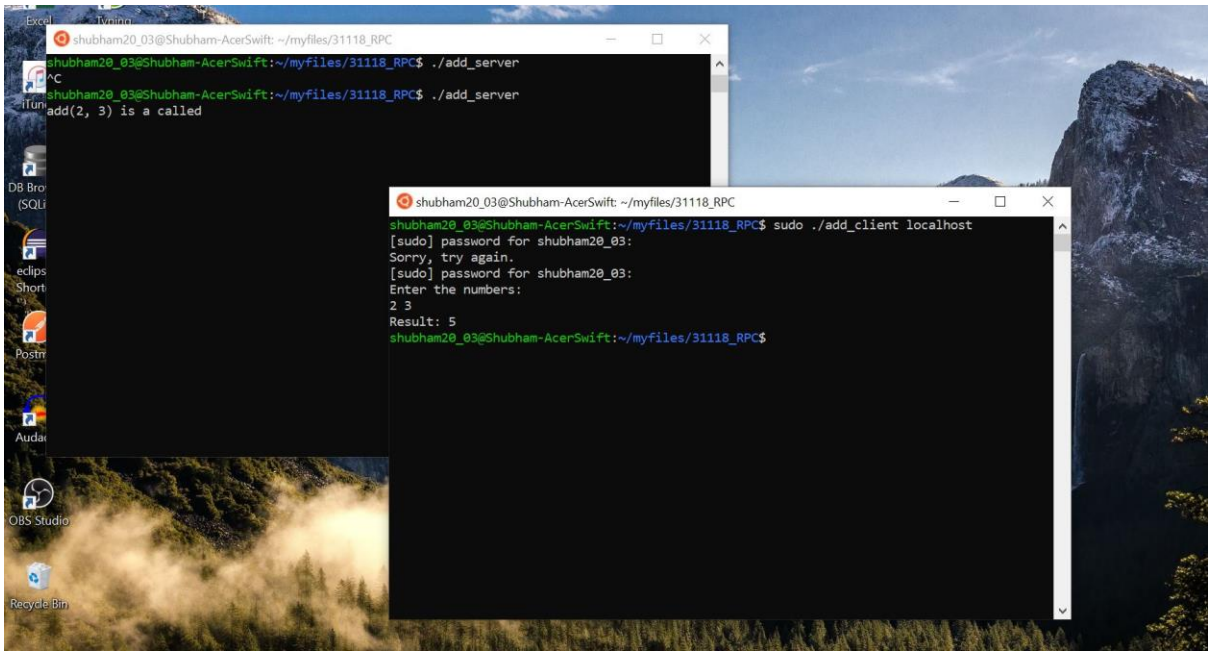
Server:

```
/*
 * Please do not edit this file.
 * It was generated using rpcgen.
 */

#include "add.h"
#include <stdio.h>
#include <stdlib.h>
#include <rpc/pmap_clnt.h>
#include <string.h>
#include <memory.h>
#include <sys/socket.h>
#include <netinet/in.h>

#ifndef SIG_PF
#define SIG_PF void(*)(int)
#endif

static void
add_prog_1(struct svc_req *rqstp, register SVCXPRT *transp)
{
	union {
		nums add_1_arg;
	} argument;
	char *result;
	xdrproc_t _xdr_argument, _xdr_result;
	char *(*local)(char *, struct svc_req *);

	switch (rqstp->rq_proc) {
	case NULLPROC:
```

```
		(void) svc_sendreply (transp, (xdrproc_t) xdr_void, (char *)NULL);
		return;

	case add:
		_xdr_argument = (xdrproc_t) xdr_nums;
		_xdr_result = (xdrproc_t) xdr_int;
		local = (char *(*)(char *, struct svc_req *)) add_1_svc;
		break;

	default:
		svcerr_noproc (transp);
		return;
	}
	memset ((char *)&argument, 0, sizeof (argument));
	if (!svc_getargs (transp, (xdrproc_t) _xdr_argument, (caddr_t) &argument)) {
		svcerr_decode (transp);
		return;
	}
	result = (*local)((char *)&argument, rqstp);
	if (result != NULL && !svc_sendreply(transp, (xdrproc_t) _xdr_result, result)) {
		svcerr_systemerr (transp);
	}
	if (!svc_freeargs (transp, (xdrproc_t) _xdr_argument, (caddr_t) &argument)) {
		fprintf (stderr, "%s", "unable to free arguments");
		exit (1);
	}
	return;
}

int
main (int argc, char **argv)
```

```
{
        register SVCXPRT *transp;

        pmap_unset (ADD_PROG, ADD_VERS);

        transp = svcudp_create(RPC_ANYSOCK);
        if (transp == NULL) {
                fprintf (stderr, "%s", "cannot create udp service.");
                exit(1);
        }
        if (!svc_register(transp, ADD_PROG, ADD_VERS, add_prog_1, IPPROTO_UDP)) {
                fprintf (stderr, "%s", "unable to register (ADD_PROG, ADD_VERS, udp).");
                exit(1);
        }

        transp = svctcp_create(RPC_ANYSOCK, 0, 0);
        if (transp == NULL) {
                fprintf (stderr, "%s", "cannot create tcp service.");
                exit(1);
        }
        if (!svc_register(transp, ADD_PROG, ADD_VERS, add_prog_1, IPPROTO_TCP)) {
                fprintf (stderr, "%s", "unable to register (ADD_PROG, ADD_VERS, tcp).");
                exit(1);
        }

        svc_run ();
        fprintf (stderr, "%s", "svc_run returned");
        exit (1);
        /* NOTREACHED */

}
```

Output:


shubham20_03@Shubham-AcerSwift: ~/myfiles/31118_RPC
shubham20_03@Shubham-AcerSwift:~/myfiles/31118_RPC$ ./add_server
^C
shubham20_03@Shubham-AcerSwift:~/myfiles/31118_RPC$ ./add_server
add(2, 3) is a called
shubham20_03@Shubham-AcerSwift: ~/myfiles/31118_RPC
shubham20_03@Shubham-AcerSwift:~/myfiles/31118_RPC$ sudo ./add_client localhost
[sudo] password for shubham20_03:
Sorry, try again.
[sudo] password for shubham20_03:
Enter the numbers:
2 3
Result: 5
shubham20_03@Shubham-AcerSwift:~/myfiles/31118_RPC$
OBS Studio
Recycle Bin