```
/* Protocol Definition file: box.x */

/* the input box dimensions */

struct dimensions
{
        float length;
        float width;
        float height;
};

/* returned results */

struct box_results
{
        float volume;
        float surface;
};


program RPC_BOX
{
        /* define version 1 of our remote program and its procedures */

        version BOXVERSION1
        {
                box_results BOX_CALC(dimensions) = 1;
        } = 1;
} = 0x40049453;
/*
Note the program number is in the temporary range 0x40000000 - 0x5FFFFFFF

The other ranges are:
0x00000000 - 0x1FFFFFFF         Administered by Sun Mircrosystems
0x20000000 - 0x3FFFFFFF         For site managers
0x40000000 - 0x5FFFFFFF         Transient uses
0x60000000 - 0xFFFFFFFF         Reserved for future use
*/
```

```
/* box_funcs.c
   server part of our rpc demo

   compile with other rpcgen-ed files
   cc box_funcs.c box_svc.c box_xdr.c -o box_svc
*/
#include"box.h"


/* some worker procedures */
float surface(float l, float w, float h)
{
        return 2*(l*w + l*h + w*h);
}

float volume(float l, float w, float h)
{
        return l*w*h;
}

/* the remote procedure */
box_results * box_calc_1_svc( dimensions *dims, struct svc_req *rqstp)
{
        static box_results res;
        /* this res structure needs to be static as we return a pointer to it*/

        /* fill in the bits */
        res.surface=surface(dims->length, dims->width, dims->height);
        res.volume=volume(dims->length, dims->width, dims->height);


        /* ship off the result */
        return &res;
}
```

```c
/* box_main.c
   does calculations about boxes using remote procedure calls

   this is the client code
   compile it with the rpcgen-ed code

   cc box_main.c box_clnt.c box_xdr.c -o box_client
*/
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include"box.h"

/* print out the box details */
void print_box(float l, float w, float h, float sa, float v)
{

        printf("A box of dimensions %f x %f x %f\n",l,w,h);
        printf("Has a surface area of %f\n",sa);
        printf("and a volume of %f\n",v);

        return;

}


int main(int ac, char**av)
{
        CLIENT *c_handle;
        dimensions dims;
        box_results *res;

        /* everything comes off the command line so check its there */
        if( ac != 5 )
        {
                fprintf(stderr,"Usage: %s remote_host length width height\n",av[0]);
                exit(EXIT_FAILURE);
        }


        /* grab the dims and shove them in the struct */
        dims.length=atof(av[2]);
        dims.width=atof(av[3]);
        dims.height=atof(av[4]);

        /* create a connection handle to make rpc's */
        c_handle=clnt_create(av[1],RPC_BOX,BOXVERSION1,"udp");
        if(c_handle==NULL)
        {
                clnt_pcreateerror(av[1]);
                exit(EXIT_FAILURE);
        }

        /* make the rpc */
        res=box_calc_1(&dims,c_handle);

        if(res==NULL)
        {
                clnt_perror(c_handle,av[1]);
                exit(EXIT_FAILURE);
        }
```

```c
        /* print out the results */
        print_box(dims.length,dims.width,dims.height,res->surface,res->volume);

        /* free up the RPC handle */
        clnt_destroy(c_handle);

        return 0;
}
```