6.) Write a Program to implement Remote Procedure Call.

```
/*
* rdate.c client program for remote date program
*/
#include <stdio.h>
#include <rpc/rpc.h> /* standard RPC include file */
#include "date.h" /* this file is generated by rpcgen */
main(int argc, char *argv[])
{
CLIENT *cl; /* RPC handle */
char *server;
long *lresult; /* return value from bin_date_1() */
char **sresult; /* return value from str_date_1() */
if (argc != 2) {
fprintf(stderr, "usage: %s hostname\n", argv[0]);
exit(1);
}
server = argv[1];
/*
* Create client handle
*/
if ((cl = clnt_create(server, DATE_PROG, DATE_VERS, "udp")) == NULL) {
/*
* can’t establish connection with server
*/
clnt_pcreateerror(server);
exit(2);
}
/*
* First call the remote procedure "bin_date".
*/
if ( (lresult = bin_date_1(NULL, cl)) == NULL) {
clnt_perror(cl, server);
exit(3);
}
printf("time on host %s = %ld\n",server, *lresult);
/*


* Now call the remote procedure str_date
*/
if ( (sresult = str_date_1(lresult, cl)) == NULL) {
clnt_perror(cl, server);
exit(4);
}
printf("time on host %s = %s", server, *sresult);
clnt_destroy(cl); /* done with the handle */
exit(0);
}
```

```
/*
* date.x Specification of the remote date and time server
*/
/*
* Define two procedures
* bin_date_1() returns the binary date and time (no arguments)
* str_date_1() takes a binary time and returns a string
*
*/
program DATE_PROG {
version DATE_VERS {
long BIN_DATE(void) = 1; /* procedure number = 1 */
string STR_DATE(long) = 2; /* procedure number = 2 */
} = 1; /* version number = 1 */
} = 0x31234567;
```

```
/*
* dateproc.c remote procedures; called by server stub
*/
#include <time.h>
#include <rpc/rpc.h> /* standard RPC include file */
#include "date.h" /* this file is generated by rpcgen */
/*
* Return the binary date and time
*/
long *bin_date_1_svc(void *arg, struct svc_req *s)
{
static long timeval; /* must be static */
timeval = time((long *) 0);
return(&timeval);
}
/*
* Convert a binary time and return a human readable string
*/
char **str_date_1_svc(long *bintime, struct svc_req *s)
{
static char *ptr; /* must be static */
ptr = ctime((const time_t *)bintime); /* convert to local time */
return(&ptr);
}
```