abc.x

```
/****
    Przyklad pliku RPCGEN
    Obliczanie sumy, roznicy i iloczynu dwoch liczb calkowitych
****/

/************************************
    Wywolanie procedury odleglej dopuszcza tylko jeden
    argument wywolania i jeden zwracany wynik.
    Oba musza byc podane w formie wskaznikow

    Dlatego nalezy definiowac odpowiednie struktury
************************************/
struct wejscie {
    int x1;
    int x2;
};
struct wyjscie {
    int suma;
    int roznica;
    int iloczyn;
};
             /* definicja programu i jego wersji */

program TESTOWY{
    version PROBNA{
                        /* definicja procedury nr 1 */
    wyjscie OBLICZENIA( wejscie) = 1;
    } = 1;              /* nr wersji */
} = 0x21000000;         /* nr programu */
```

abc.h

```
/*
 * Please do not edit this file.
 * It was generated using rpcgen.  */
#ifndef _ABC_H_RPCGEN
#define _ABC_H_RPCGEN
#include <rpc/rpc.h>
#ifdef __cplusplus
extern "C" {
#endif
struct wejscie {
    int x1;
    int x2;
};
typedef struct wejscie wejscie;
struct wyjscie {
    int suma;
    int roznica;
    int iloczyn;
};
typedef struct wyjscie wyjscie;
#define TESTOWY 0x21000000
#define PROBNA 1
#if defined(__STDC__) || defined(__cplusplus)
#define OBLICZENIA 1
extern  wyjscie * obliczenia_1(wejscie *, CLIENT *);
extern  wyjscie * obliczenia_1_svc(wejscie *, struct svc_req *);
extern int testowy_1_freeresult (SVCXPRT *, xdrproc_t, caddr_t);
#else /* K&R C */
#define OBLICZENIA 1
extern  wyjscie * obliczenia_1();
extern  wyjscie * obliczenia_1_svc();
extern int testowy_1_freeresult ();
#endif /* K&R C */
/* the xdr functions */
#if defined(__STDC__) || defined(__cplusplus)
extern  bool_t xdr_wejscie (XDR *, wejscie*);
extern  bool_t xdr_wyjscie (XDR *, wyjscie*);
#else /* K&R C */
extern bool_t xdr_wejscie ();
extern bool_t xdr_wyjscie ();
#endif /* K&R C */
#ifdef __cplusplus
}
#endif
#endif /* !_ABC_H_RPCGEN */
```

abc_xdr.c

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

#include "abc.h"

bool_t
xdr_wejscie (XDR *xdrs, wejscie *objp)
{
    register int32_t *buf;

     if (!xdr_int (xdrs, &objp->x1))
          return FALSE;
     if (!xdr_int (xdrs, &objp->x2))
          return FALSE;
    return TRUE;
}

bool_t
xdr_wyjscie (XDR *xdrs, wyjscie *objp)
{
    register int32_t *buf;

     if (!xdr_int (xdrs, &objp->suma))
          return FALSE;
     if (!xdr_int (xdrs, &objp->roznica))
          return FALSE;
     if (!xdr_int (xdrs, &objp->iloczyn))
          return FALSE;
    return TRUE;
}
```

abc_clnt.c

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

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

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

wyjscie *
obliczenia_1(wejscie *argp, CLIENT *clnt)
{
    static wyjscie clnt_res;

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

abc_client.c (część 1)

```
/****
    Program glowny klienta
****/
/*
 * 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 "abc.h"
void
testowy_1(char *host, int liczba1, int liczba2)
{
    CLIENT *clnt;
    wyjscie  *result_1;
    wejscie  obliczenia_1_arg;

#ifndef  DEBUG
    clnt = clnt_create (host, TESTOWY, PROBNA, "udp");
    if (clnt == NULL) {
         clnt_pcreateerror (host);
         exit (1);
    }
#endif   /* DEBUG */

    obliczenia_1_arg.x1 = liczba1;
    obliczenia_1_arg.x2 = liczba2;

    result_1 = obliczenia_1(&obliczenia_1_arg, clnt);
    if (result_1 == (wyjscie *) NULL) {
         clnt_perror (clnt, "call failed");
    }

    printf("\nWprowadzone dane:\t%d\t%d\n", liczba1, liczba2);
    printf("\nUzyskane wyniki:\n");
    printf("\tsuma:\t\t%d\n\troznica:\t%d\n\tiloczyn:\t%d\n\n",
         result_1->suma, result_1->roznica, result_1->iloczyn);

#ifndef  DEBUG
    clnt_destroy (clnt);
#endif    /* DEBUG */
}
```

abc_client.c (część 2)

```
/****
    Procedura klienta w pliku klient.c
****/

int
main (int argc, char *argv[])
{
    char *host;

    if (argc < 4) {
         printf ("usage: %s server_host  .....\n", argv[0]);
         exit (1);
    }
    host = argv[1];
    testowy_1 (host, atoi(argv[2]), atoi(argv[3]));
exit (0);
}
```

abc_server.c

```
/****
    Definicja procedury RPC po stronie serwera
****/
/*
 * 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 "abc.h"
wyjscie *
obliczenia_1_svc(wejscie *argp, struct svc_req *rqstp)
{
    static wyjscie  result;
    /*
     * insert server code here
     */
    result.suma    = argp->x1 + argp->x2;
    result.roznica = argp->x1 - argp->x2;
    result.iloczyn = argp->x1 * argp->x2;
    return &result;
}
```