

## **add.x**

```
/*combine the arguments to be passed to the server side in a structure*/
struct numbers{
    int a;
    int b;
};
program ADD_PROG{
version ADD_VERS{
    int add(numbers)=1;
} =1;
}=0x4562877;
```

## **add\_client.c**

```
#include "add.h"

void add_prog_1(char *host, int x, int y) {
    CLIENT *clnt;
    int *result_1;
    numbers add_1_arg;

#ifdef DEBUG
    clnt = clnt_create(host, ADD_PROG, ADD_VERS, "udp");
    if (clnt == NULL) {
        clnt_pcreateerror(host);
        exit(1);
    }
#endif /* DEBUG */

    add_1_arg.a = x;
    add_1_arg.b = y;
    result_1 = add_1(&add_1_arg, clnt);

    if (result_1 == (int *)NULL) {
        clnt_perror(clnt, "call failed");
    } else {
        printf("Result: %d\n", *result_1);
    }

#ifdef DEBUG
    clnt_destroy(clnt);
#endif /* DEBUG */
}

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

    if (argc < 4) {
        printf("usage: %s server_host\n", argv[0]);
        exit(1);
    }
}
```

```

    host = argv[1];
    add_prog_1(host, atoi(argv[2]), atoi(argv[3]));

    exit(0);
}

```

### **add\_server.c**

```

#include "add.h"

int *add_1_svc(numbers *argp, struct svc_req *rqstp) {
    static int result;
    printf("add(%d, %d) is called\n", argp->a, argp->b);
    result = argp->a + argp->b;
    return &result;
}

```

### **sub.x**

```

/*combine the arguments to be passed to the server side in a structure*/

struct numbers{

int a;

int b;

};

program SUB_PROG{

version SUB_VERS{

int sub(numbers)=1;

}=1;

}=0x4562877;

```

### **sub\_client.c**

```

#include "sub.h"

void sub_prog_1(char *host, int x, int y) {
    CLIENT *clnt;
    int *result_1;
    numbers sub_1_arg;

#ifdef DEBUG
    clnt = clnt_create(host, SUB_PROG, SUB_VERS, "udp");
    if (clnt == NULL) {
        clnt_pcreateerror(host);
    }

```

```

        exit(1);
    }
#endif /* DEBUG */

    sub_1_arg.a = x;
    sub_1_arg.b = y;
    result_1 = sub_1(&sub_1_arg, clnt);

    if (result_1 == (int *)NULL) {
        clnt_perror(clnt, "call failed");
    } else {
        printf("Result: %d\n", *result_1);
    }

#ifdef DEBUG
    clnt_destroy(clnt);
#endif /* DEBUG */
}

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

    if (argc < 4) {
        printf("usage: %s server_host\n", argv[0]);
        exit(1);
    }

    host = argv[1];
    sub_prog_1(host, atoi(argv[2]), atoi(argv[3]));

    exit(0);
}

```

### **sub\_server.c**

```

#include "sub.h"

int *sub_1_svc(numbers *argp, struct svc_req *rqstp) {
    static int result;
    printf("sub(%d, %d) is called\n", argp->a, argp->b);
    result = argp->a - argp->b;
    return &result;
}

```