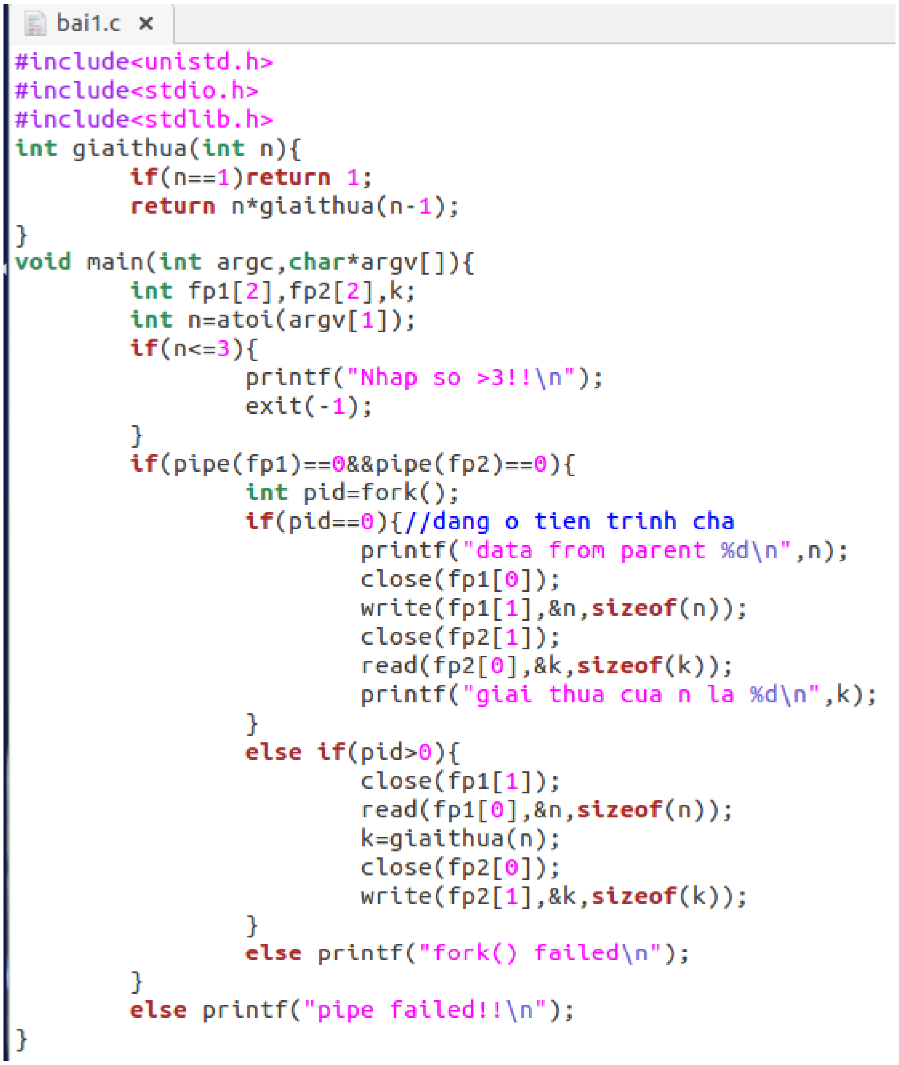
Huỳnh Tấn Dương

3122410061

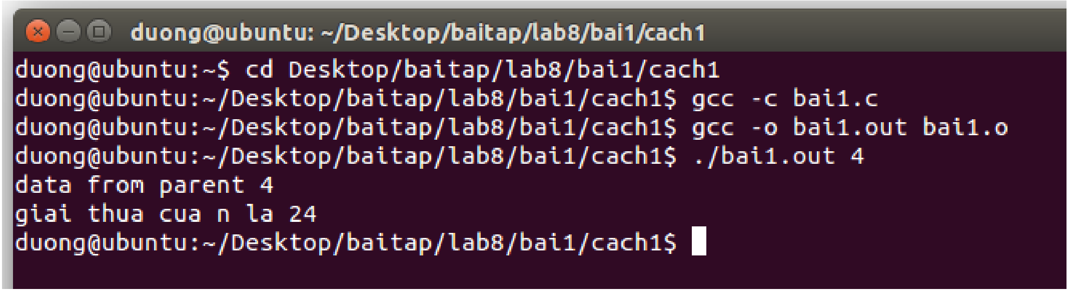
Hoàn thành 2/2

Bài 1/

Cách 1

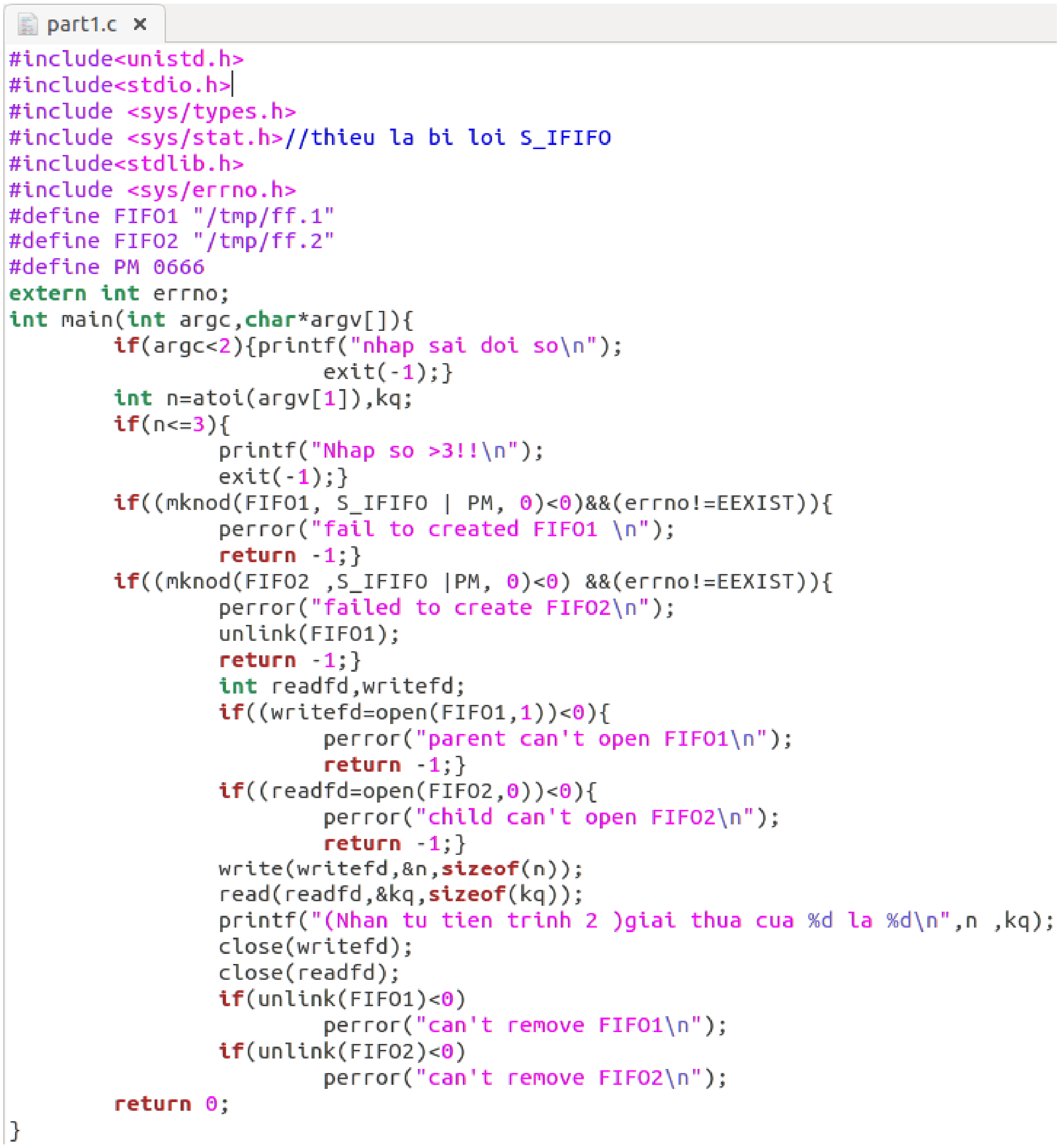


```
#include<unistd.h>
#include<stdio.h>
#include<stdlib.h>
int giaithua(int n){
        if(n==1)return 1;
        return n*giaithua(n-1);
}
void main(int argc,char*argv[]){
        int fp1[2],fp2[2],k;
        int n=atoi(argv[1]);
        if(n<=3){
                printf("Nhap so >3!!\n");
                exit(-1);
        }
        if(pipe(fp1)==0&&pipe(fp2)==0){
                int pid=fork();
                if(pid==0){//dang o tien trinh cha
                        printf("data from parent %d\n",n);
                        close(fp1[0]);
                        write(fp1[1],&n,sizeof(n));
                        close(fp2[1]);
                        read(fp2[0],&k,sizeof(k));
                        printf("giai thua cua n la %d\n",k);
                }
                else if(pid>0){
                        close(fp1[1]);
                        read(fp1[0],&n,sizeof(n));
                        k=giaithua(n);
                        close(fp2[0]);
                        write(fp2[1],&k,sizeof(k));
                }
                else printf("fork() failed\n");
        }
        else printf("pipe failed!!\n");
}
```

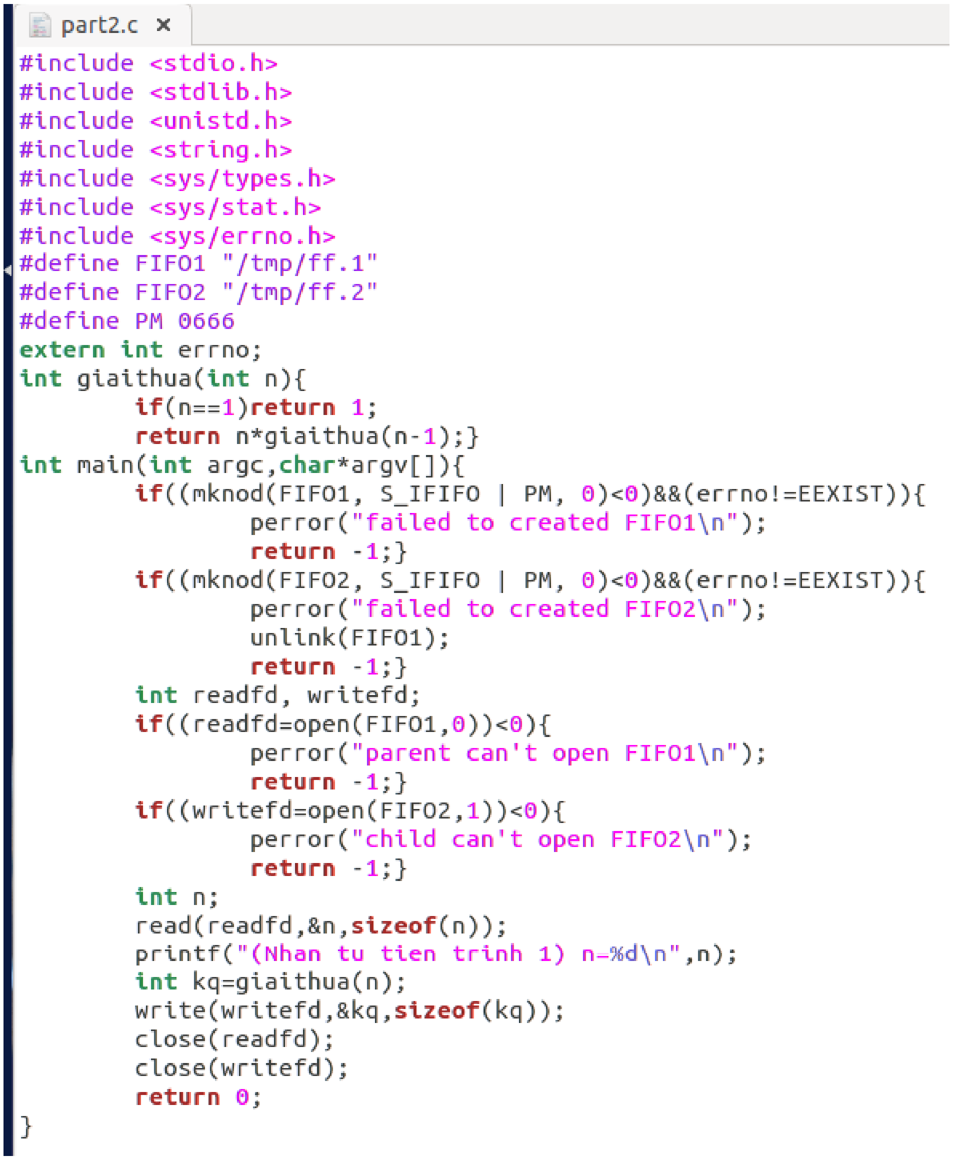


```
duong@ubuntu:~$ cd Desktop/baitap/lab8/bai1/cach1
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach1$ gcc -c bai1.c
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach1$ gcc -o bai1.out bai1.o
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach1$ ./bai1.out 4
data from parent 4
giai thua cua n la 24
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach1$ 
```

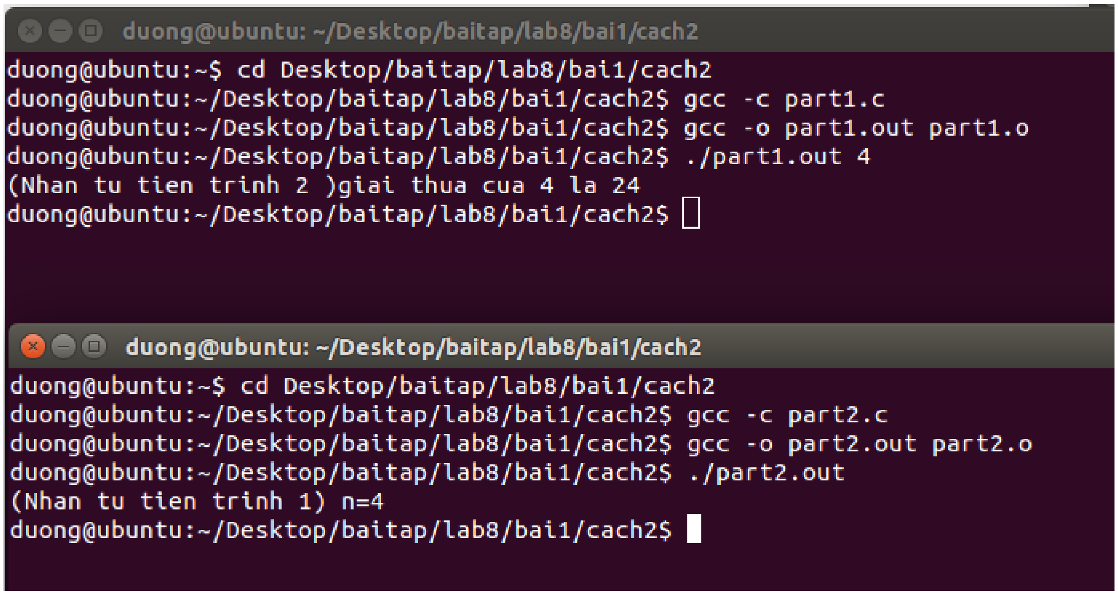
Cách 2



```
#include<unistd.h>
#include<stdio.h>
#include <sys/types.h>
#include <sys/stat.h>//thieu la bi loi S_IFIFO
#include<stdlib.h>
#include <sys/errno.h>
#define FIFO1 "/tmp/ff.1"
#define FIFO2 "/tmp/ff.2"
#define PM 0666
extern int errno;
int main(int argc,char*argv[]){
        if(argc<2){printf("nhap sai doi so\n");
                        exit(-1);}
        int n=atoi(argv[1]),kq;
        if(n<=3){
                printf("Nhap so >3!!\n");
                exit(-1);}
        if((mknod(FIFO1, S_IFIFO | PM, 0)<0)&&(errno!=EEXIST)){
                perror("fail to created FIFO1 \n");
                return -1;}
        if((mknod(FIFO2 ,S_IFIFO |PM, 0)<0) &&(errno!=EEXIST)){
                perror("failed to create FIFO2\n");
                unlink(FIFO1);
                return -1;}
                int readfd,writefd;
                if((writefd=open(FIFO1,1))<0){
                        perror("parent can't open FIFO1\n");
                        return -1;}
                if((readfd=open(FIFO2,0))<0){
                        perror("child can't open FIFO2\n");
                        return -1;}
                write(writefd,&n,sizeof(n));
                read(readfd,&kq,sizeof(kq));
                printf("(Nhan tu tien trinh 2 )giai thua cua %d la %d\n",n ,kq);
                close(writefd);
                close(readfd);
                if(unlink(FIFO1)<0)
                        perror("can't remove FIFO1\n");
                if(unlink(FIFO2)<0)
                        perror("can't remove FIFO2\n");
        return 0;
}
```



```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/errno.h>
#define FIFO1 "/tmp/ff.1"
#define FIFO2 "/tmp/ff.2"
#define PM 0666
extern int errno;
int giaithua(int n){
        if(n==1)return 1;
        return n*giaithua(n-1);}
int main(int argc,char*argv[]){
        if((mknod(FIFO1, S_IFIFO | PM, 0)<0)&&(errno!=EEXIST)){
                perror("failed to created FIFO1\n");
                return -1;}
        if((mknod(FIFO2, S_IFIFO | PM, 0)<0)&&(errno!=EEXIST)){
                perror("failed to created FIFO2\n");
                unlink(FIFO1);
                return -1;}
        int readfd, writefd;
        if((readfd=open(FIFO1,0))<0){
                perror("parent can't open FIFO1\n");
                return -1;}
        if((writefd=open(FIFO2,1))<0){
                perror("child can't open FIFO2\n");
                return -1;}
        int n;
        read(readfd,&n,sizeof(n));
        printf("(Nhan tu tien trinh 1) n=%d\n",n);
        int kq=giaithua(n);
        write(writefd,&kq,sizeof(kq));
        close(readfd);
        close(writefd);
        return 0;
}
```

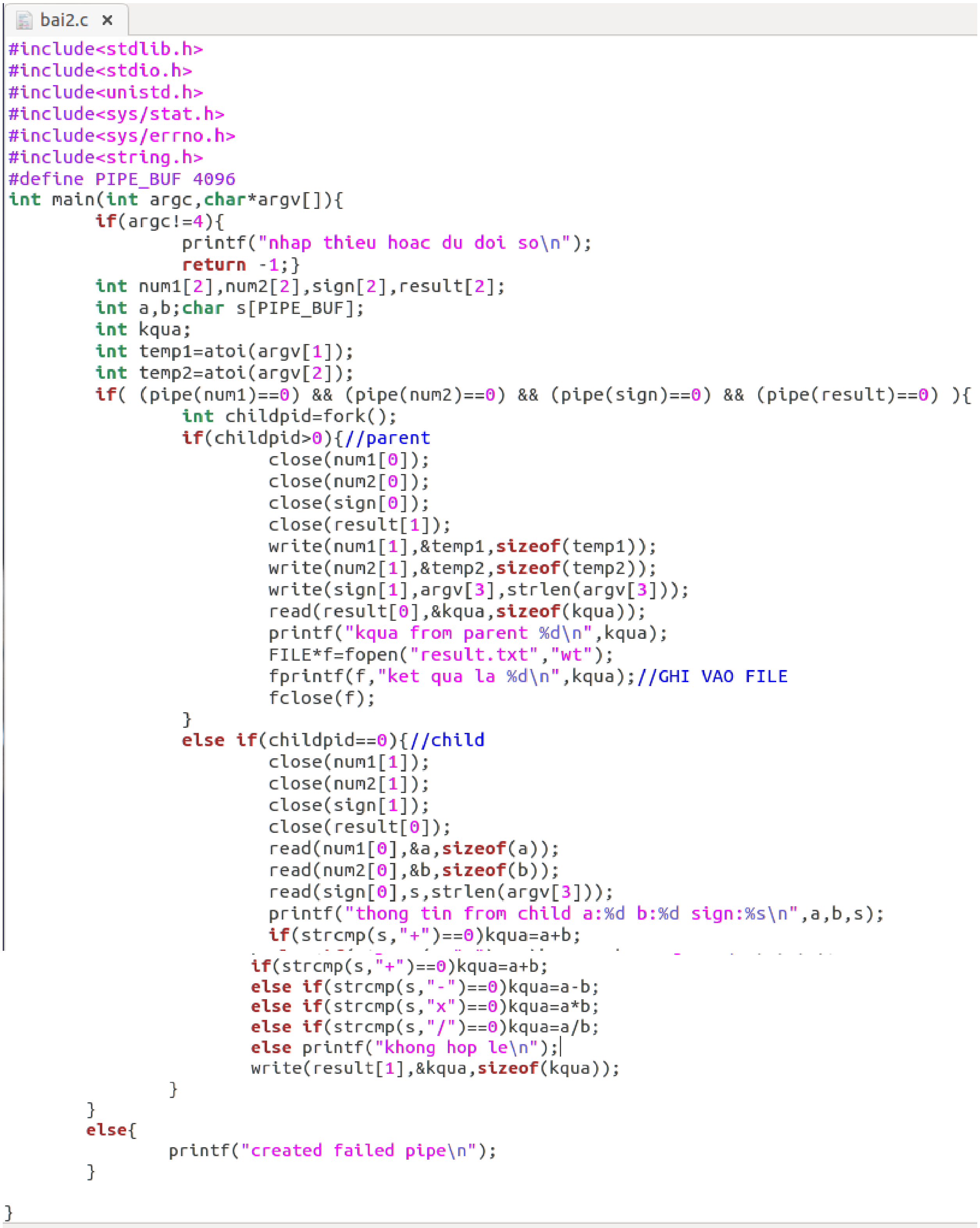

duong@ubuntu: ~/Desktop/baitap/lab8/bai1/cach2
duong@ubuntu: ~/Desktop/baitap/lab8/bai1/cach2

```
duong@ubuntu:~$ cd Desktop/baitap/lab8/bai1/cach2
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach2$ gcc -c part1.c
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach2$ gcc -o part1.out part1.o
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach2$ ./part1.out 4
(Nhan tu tien trinh 2 )giai thua cua 4 la 24
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach2$
```

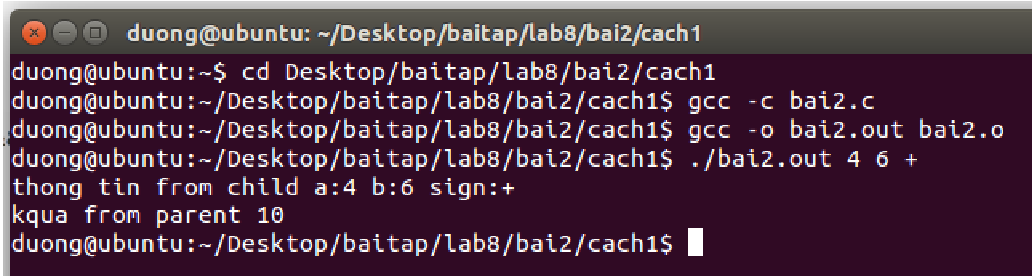
```
duong@ubuntu:~$ cd Desktop/baitap/lab8/bai1/cach2
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach2$ gcc -c part2.c
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach2$ gcc -o part2.out part2.o
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach2$ ./part2.out
(Nhan tu tien trinh 1) n=4
duong@ubuntu:~/Desktop/baitap/lab8/bai1/cach2$
```

Bài 2/

Cách 1

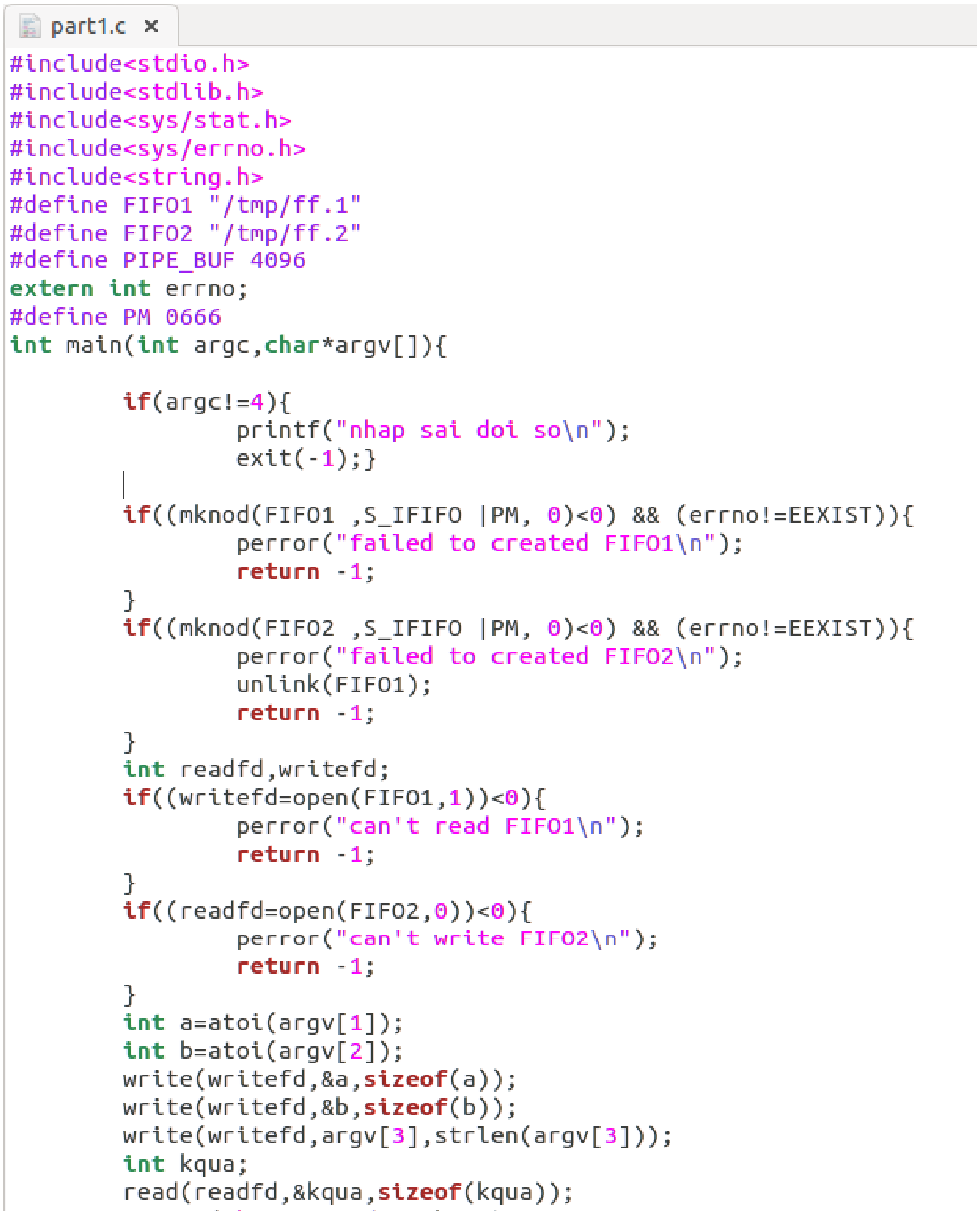


```c
#include<stdlib.h>
#include<stdio.h>
#include<unistd.h>
#include<sys/stat.h>
#include<sys/errno.h>
#include<string.h>
#define PIPE_BUF 4096
int main(int argc,char*argv[]){
        if(argc!=4){
                printf("nhap thieu hoac du doi so\n");
                return -1;}
        int num1[2],num2[2],sign[2],result[2];
        int a,b;char s[PIPE_BUF];
        int kqua;
        int temp1=atoi(argv[1]);
        int temp2=atoi(argv[2]);
        if( (pipe(num1)==0) && (pipe(num2)==0) && (pipe(sign)==0) && (pipe(result)==0) ){
                int childpid=fork();
                if(childpid>0){//parent
                        close(num1[0]);
                        close(num2[0]);
                        close(sign[0]);
                        close(result[1]);
                        write(num1[1],&temp1,sizeof(temp1));
                        write(num2[1],&temp2,sizeof(temp2));
                        write(sign[1],argv[3],strlen(argv[3]));
                        read(result[0],&kqua,sizeof(kqua));
                        printf("kqua from parent %d\n",kqua);
                        FILE*f=fopen("result.txt","wt");
                        fprintf(f,"ket qua la %d\n",kqua);//GHI VAO FILE
                        fclose(f);
                }
                else if(childpid==0){//child
                        close(num1[1]);
                        close(num2[1]);
                        close(sign[1]);
                        close(result[0]);
                        read(num1[0],&a,sizeof(a));
                        read(num2[0],&b,sizeof(b));
                        read(sign[0],s,strlen(argv[3]));
                        printf("thong tin from child a:%d b:%d sign:%s\n",a,b,s);
                        if(strcmp(s,"+")==0)kqua=a+b;
                        else if(strcmp(s,"-")==0)kqua=a-b;
                        else if(strcmp(s,"x")==0)kqua=a*b;
                        else if(strcmp(s,"/")==0)kqua=a/b;
                        else printf("khong hop le\n");
                        write(result[1],&kqua,sizeof(kqua));
                }
        }
        else{
                printf("created failed pipe\n");
        }

}
```

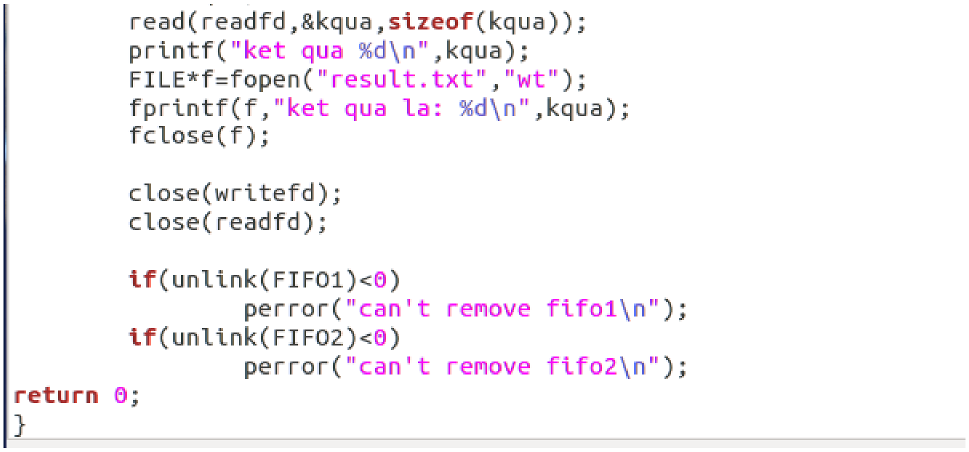


```
duong@ubuntu:~$ cd Desktop/baitap/lab8/bai2/cach1
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach1$ gcc -c bai2.c
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach1$ gcc -o bai2.out bai2.o
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach1$ ./bai2.out 4 6 +
thong tin from child a:4 b:6 sign:+
kqua from parent 10
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach1$ 
```

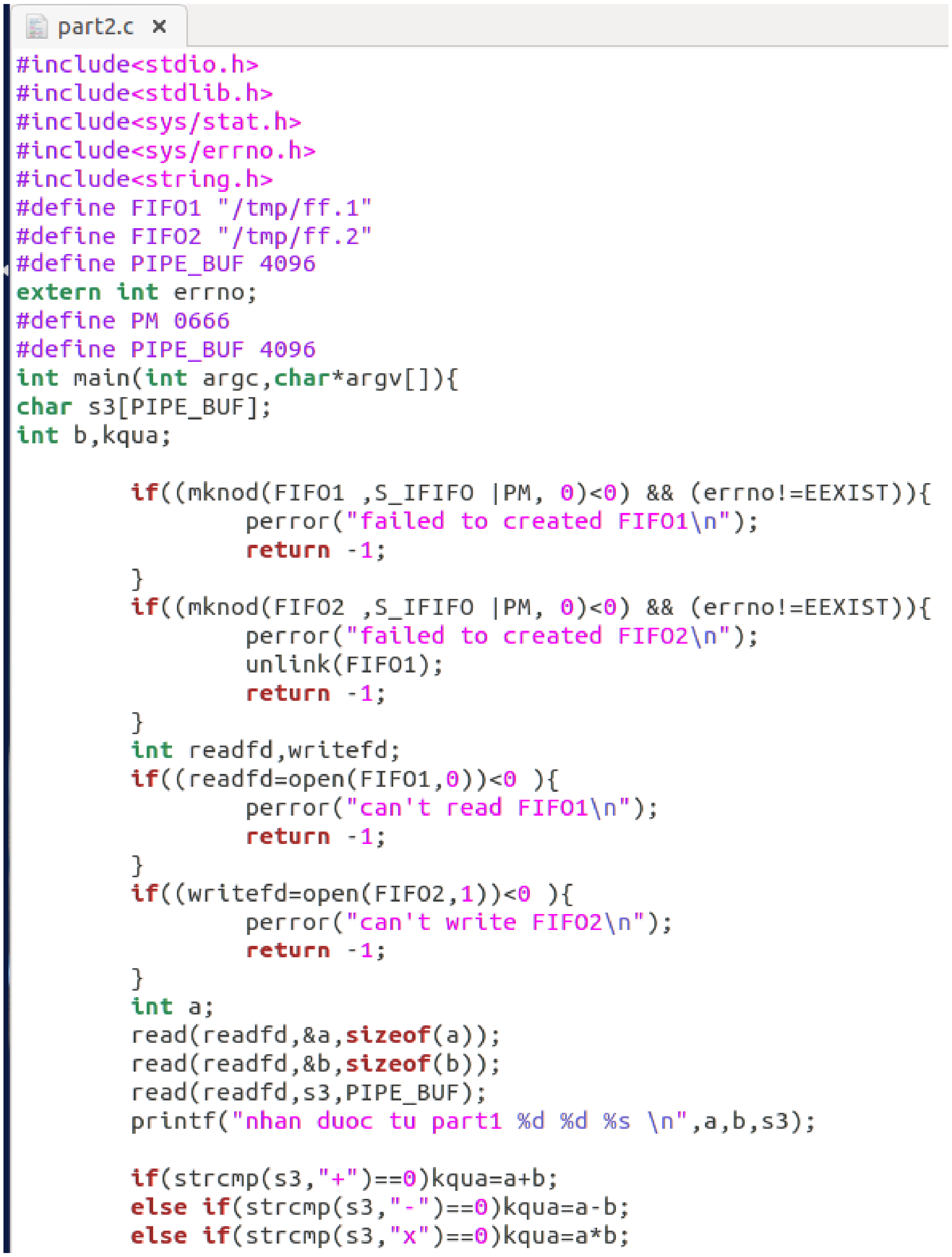
Cách 2



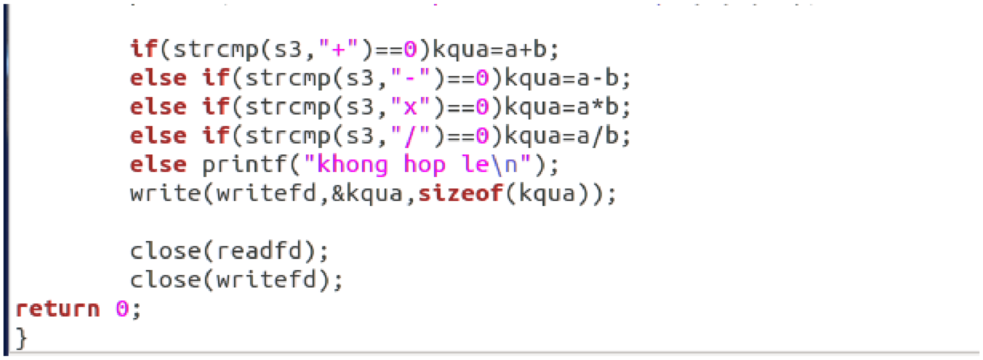
```c
#include<stdio.h>
#include<stdlib.h>
#include<sys/stat.h>
#include<sys/errno.h>
#include<string.h>
#define FIFO1 "/tmp/ff.1"
#define FIFO2 "/tmp/ff.2"
#define PIPE_BUF 4096
extern int errno;
#define PM 0666
int main(int argc,char*argv[]){

        if(argc!=4){
                printf("nhap sai doi so\n");
                exit(-1);}

        if((mknod(FIFO1 ,S_IFIFO |PM, 0)<0) && (errno!=EEXIST)){
                perror("failed to created FIFO1\n");
                return -1;
        }
        if((mknod(FIFO2 ,S_IFIFO |PM, 0)<0) && (errno!=EEXIST)){
                perror("failed to created FIFO2\n");
                unlink(FIFO1);
                return -1;
        }
        int readfd,writefd;
        if((writefd=open(FIFO1,1))<0){
                perror("can't read FIFO1\n");
                return -1;
        }
        if((readfd=open(FIFO2,0))<0){
                perror("can't write FIFO2\n");
                return -1;
        }
        int a=atoi(argv[1]);
        int b=atoi(argv[2]);
        write(writefd,&a,sizeof(a));
        write(writefd,&b,sizeof(b));
        write(writefd,argv[3],strlen(argv[3]));
        int kqua;
        read(readfd,&kqua,sizeof(kqua));
```



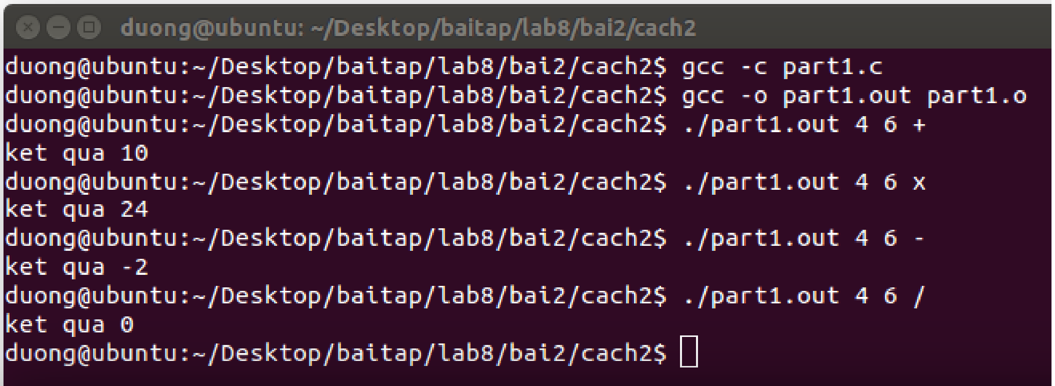
```
        read(readfd,&kqua,sizeof(kqua));
        printf("ket qua %d\n",kqua);
        FILE*f=fopen("result.txt","wt");
        fprintf(f,"ket qua la: %d\n",kqua);
        fclose(f);

        close(writefd);
        close(readfd);

        if(unlink(FIFO1)<0)
                perror("can't remove fifo1\n");
        if(unlink(FIFO2)<0)
                perror("can't remove fifo2\n");
return 0;
}
```



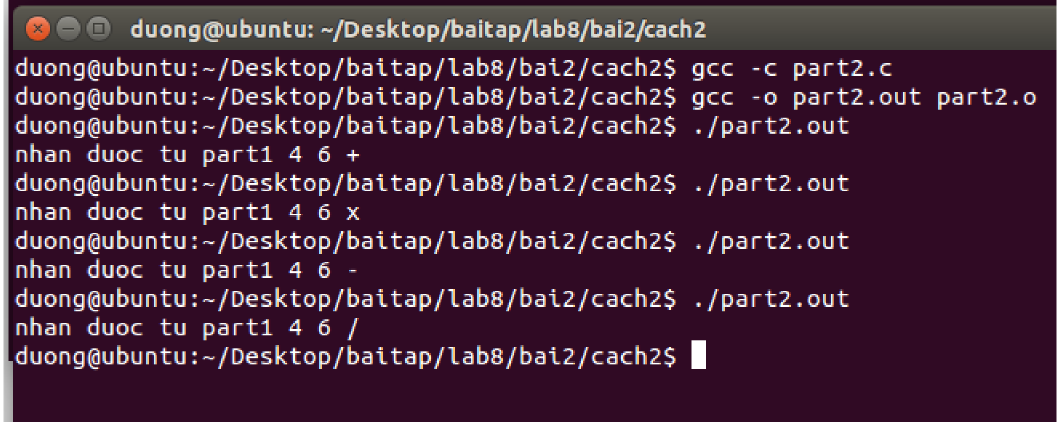
```
#include<stdio.h>
#include<stdlib.h>
#include<sys/stat.h>
#include<sys/errno.h>
#include<string.h>
#define FIFO1 "/tmp/ff.1"
#define FIFO2 "/tmp/ff.2"
#define PIPE_BUF 4096
extern int errno;
#define PM 0666
#define PIPE_BUF 4096
int main(int argc,char*argv[]){
char s3[PIPE_BUF];
int b,kqua;

        if((mknod(FIFO1 ,S_IFIFO |PM, 0)<0) && (errno!=EEXIST)){
                perror("failed to created FIFO1\n");
                return -1;
        }
        if((mknod(FIFO2 ,S_IFIFO |PM, 0)<0) && (errno!=EEXIST)){
                perror("failed to created FIFO2\n");
                unlink(FIFO1);
                return -1;
        }
        int readfd,writefd;
        if((readfd=open(FIFO1,0))<0 ){
                perror("can't read FIFO1\n");
                return -1;
        }
        if((writefd=open(FIFO2,1))<0 ){
                perror("can't write FIFO2\n");
                return -1;
        }
        int a;
        read(readfd,&a,sizeof(a));
        read(readfd,&b,sizeof(b));
        read(readfd,s3,PIPE_BUF);
        printf("nhan duoc tu part1 %d %d %s \n",a,b,s3);

        if(strcmp(s3,"+")==0)kqua=a+b;
        else if(strcmp(s3,"-")==0)kqua=a-b;
        else if(strcmp(s3,"x")==0)kqua=a*b;
```



```
        if(strcmp(s3,"+")==0)kqua=a+b;
        else if(strcmp(s3,"-")==0)kqua=a-b;
        else if(strcmp(s3,"x")==0)kqua=a*b;
        else if(strcmp(s3,"/")==0)kqua=a/b;
        else printf("khong hop le\n");
        write(writefd,&kqua,sizeof(kqua));

        close(readfd);
        close(writefd);
return 0;
}
```



```
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ gcc -c part1.c
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ gcc -o part1.out part1.o
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ ./part1.out 4 6 +
ket qua 10
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ ./part1.out 4 6 x
ket qua 24
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ ./part1.out 4 6 -
ket qua -2
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ ./part1.out 4 6 /
ket qua 0
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ 
```



```
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ gcc -c part2.c
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ gcc -o part2.out part2.o
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ ./part2.out
nhan duoc tu part1 4 6 +
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ ./part2.out
nhan duoc tu part1 4 6 x
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ ./part2.out
nhan duoc tu part1 4 6 -
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ ./part2.out
nhan duoc tu part1 4 6 /
duong@ubuntu:~/Desktop/baitap/lab8/bai2/cach2$ 
```