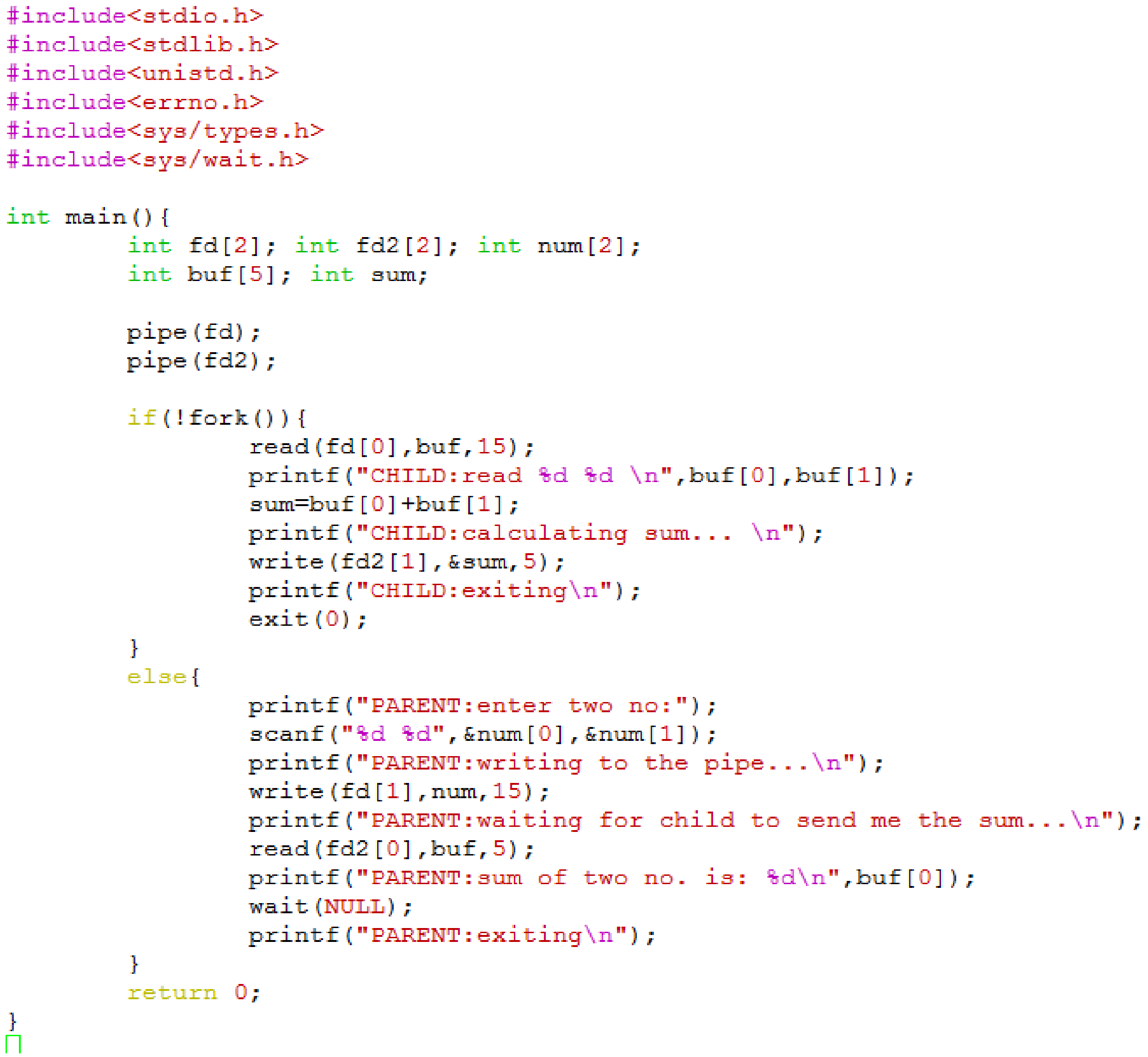
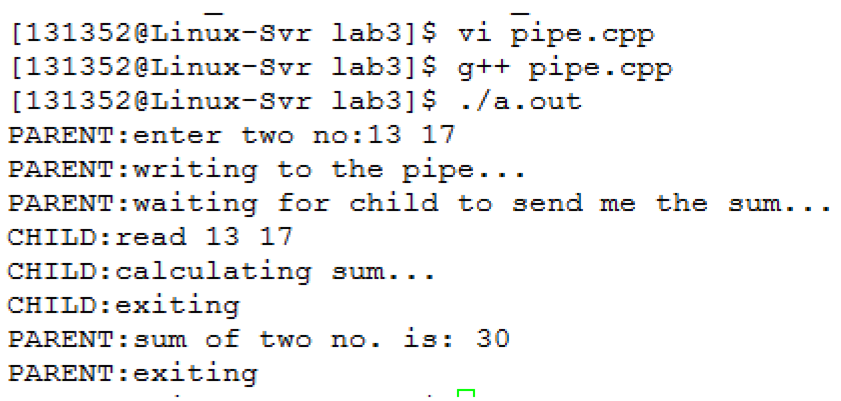
LAB-3

1. Write a C programe using the pipe that perfrom following work:
   (a) Parent: Ask two no from user and pass the number to child process.
   (b) Child: add the numbers recvied from the parent and pass the result to parent.
   (c) Parent: print the table of output received from child.

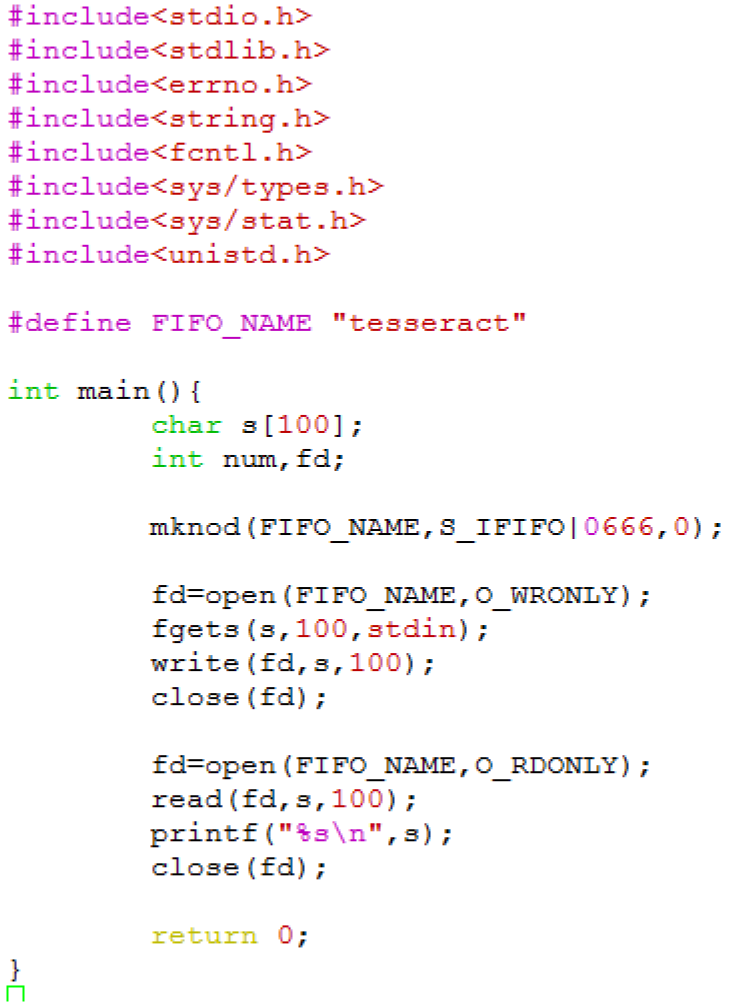


```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<errno.h>
#include<sys/types.h>
#include<sys/wait.h>

int main(){
        int fd[2]; int fd2[2]; int num[2];
        int buf[5]; int sum;

        pipe(fd);
        pipe(fd2);

        if(!fork()){
                read(fd[0],buf,15);
                printf("CHILD:read %d %d \n",buf[0],buf[1]);
                sum=buf[0]+buf[1];
                printf("CHILD:calculating sum... \n");
                write(fd2[1],&sum,5);
                printf("CHILD:exiting\n");
                exit(0);
        }
        else{
                printf("PARENT:enter two no:");
                scanf("%d %d",&num[0],&num[1]);
                printf("PARENT:writing to the pipe...\n");
                write(fd[1],num,15);
                printf("PARENT:waiting for child to send me the sum...\n");
                read(fd2[0],buf,5);
                printf("PARENT:sum of two no. is: %d\n",buf[0]);
                wait(NULL);
                printf("PARENT:exiting\n");
        }
        return 0;
}
```

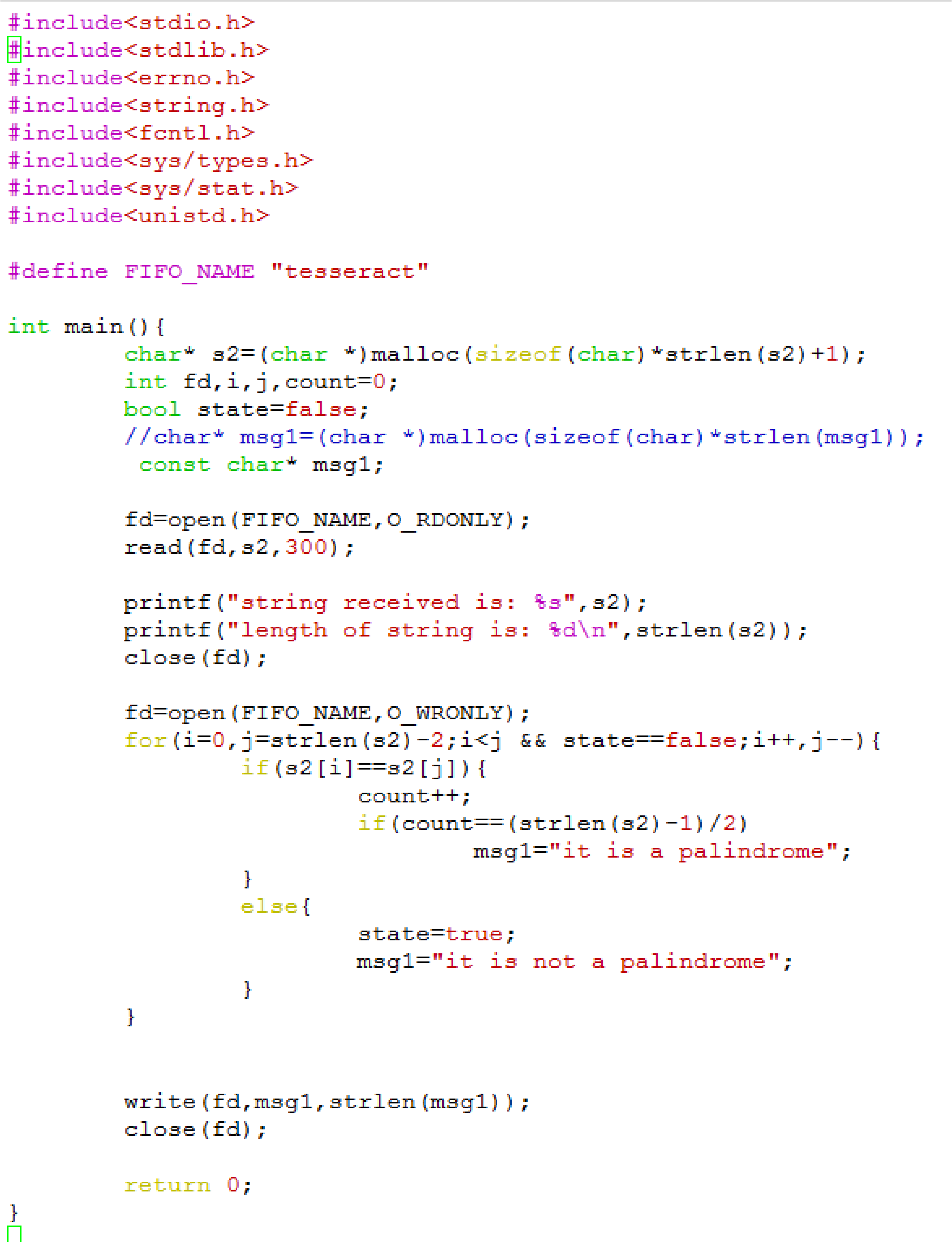


```
[131352@Linux-Svr lab3]$ vi pipe.cpp
[131352@Linux-Svr lab3]$ g++ pipe.cpp
[131352@Linux-Svr lab3]$ ./a.out
PARENT:enter two no:13 17
PARENT:writing to the pipe...
PARENT:waiting for child to send me the sum...
CHILD:read 13 17
CHILD:calculating sum...
CHILD:exiting
PARENT:sum of two no. is: 30
PARENT:exiting
```

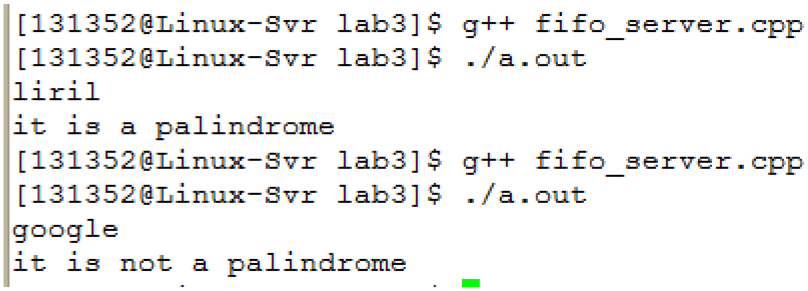
2. Write a C programe using the fifo that perfrom following work:
   (a) Parent: Ask a string from user and pass the string to child process.
   (b) Child: check the string (recvied from the parent) is palindrome or not and pass the result to parent.
   (c) Parent: print the output received from child on the screen.



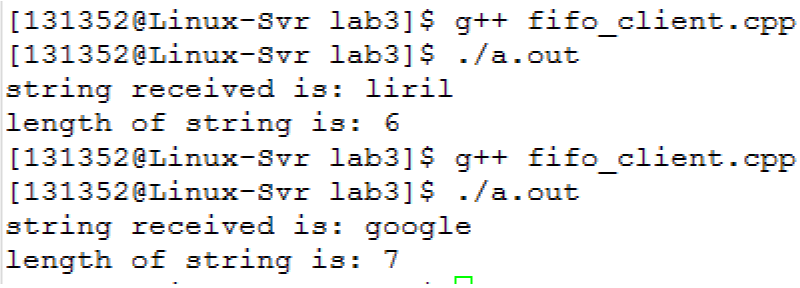
```
#include<stdio.h>
#include<stdlib.h>
#include<errno.h>
#include<string.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>

#define FIFO_NAME "tesseract"

int main(){
        char s[100];
        int num,fd;

        mknod(FIFO_NAME,S_IFIFO|0666,0);

        fd=open(FIFO_NAME,O_WRONLY);
        fgets(s,100,stdin);
        write(fd,s,100);
        close(fd);

        fd=open(FIFO_NAME,O_RDONLY);
        read(fd,s,100);
        printf("%s\n",s);
        close(fd);

        return 0;
}
```



```
#include<stdio.h>
#include<stdlib.h>
#include<errno.h>
#include<string.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>

#define FIFO_NAME "tesseract"

int main(){
        char* s2=(char *)malloc(sizeof(char)*strlen(s2)+1);
        int fd,i,j,count=0;
        bool state=false;
        //char* msg1=(char *)malloc(sizeof(char)*strlen(msg1));
         const char* msg1;

        fd=open(FIFO_NAME,O_RDONLY);
        read(fd,s2,300);

        printf("string received is: %s",s2);
        printf("length of string is: %d\n",strlen(s2));
        close(fd);

        fd=open(FIFO_NAME,O_WRONLY);
        for(i=0,j=strlen(s2)-2;i<j && state==false;i++,j--){
                if(s2[i]==s2[j]){
                        count++;
                        if(count==(strlen(s2)-1)/2)
                                msg1="it is a palindrome";
                }
                else{
                        state=true;
                        msg1="it is not a palindrome";
                }
        }


        write(fd,msg1,strlen(msg1));
        close(fd);

        return 0;
}
```



```
[131352@Linux-Svr lab3]$ g++ fifo_server.cpp
[131352@Linux-Svr lab3]$ ./a.out
liril
it is a palindrome
[131352@Linux-Svr lab3]$ g++ fifo_server.cpp
[131352@Linux-Svr lab3]$ ./a.out
google
it is not a palindrome
```



```
[131352@Linux-Svr lab3]$ g++ fifo_client.cpp
[131352@Linux-Svr lab3]$ ./a.out
string received is: liril
length of string is: 6
[131352@Linux-Svr lab3]$ g++ fifo_client.cpp
[131352@Linux-Svr lab3]$ ./a.out
string received is: google
length of string is: 7
```

LAB-4

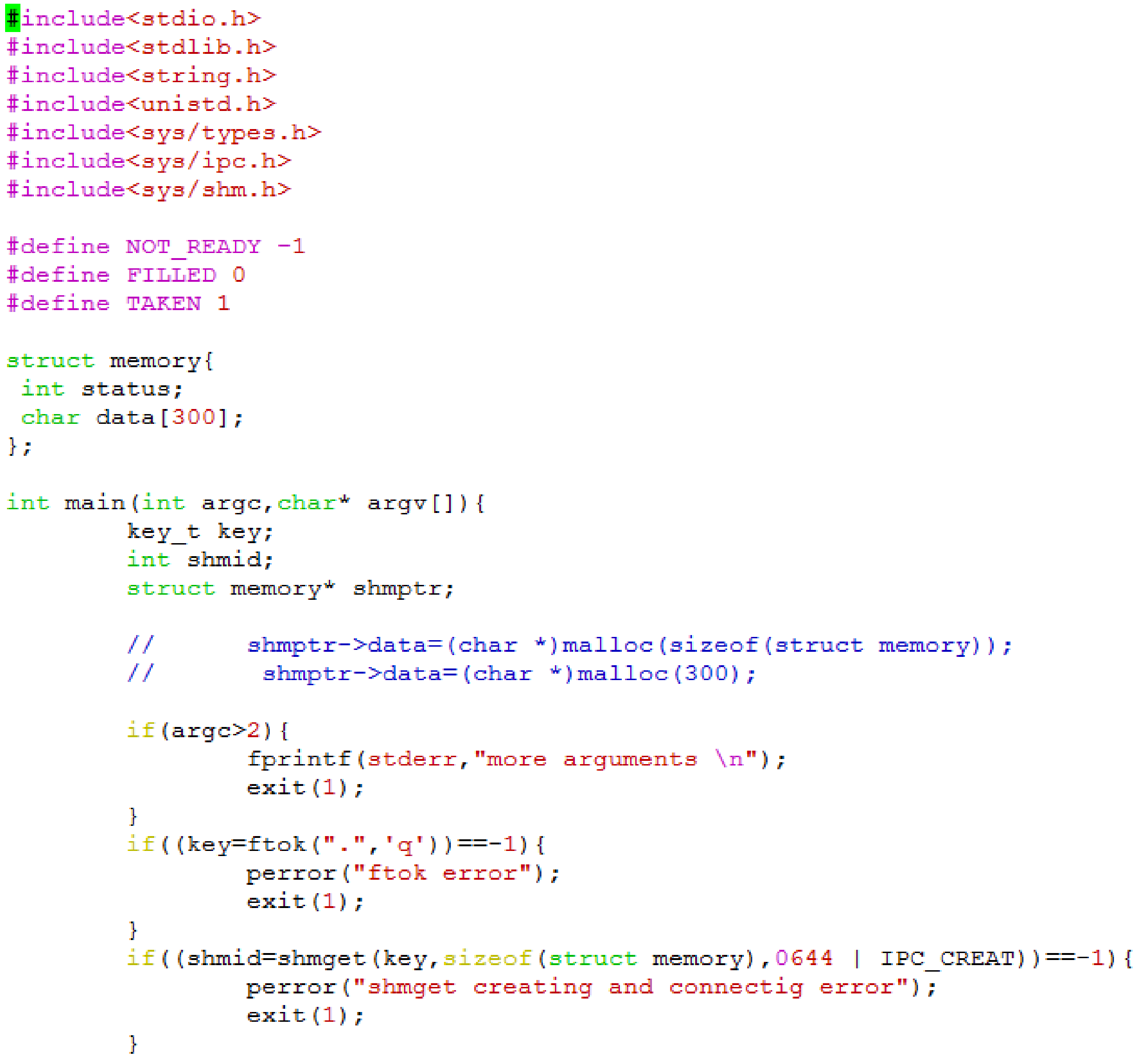
Shared Memory

Q. Write a C program using shared memory. Processes will perform following task:

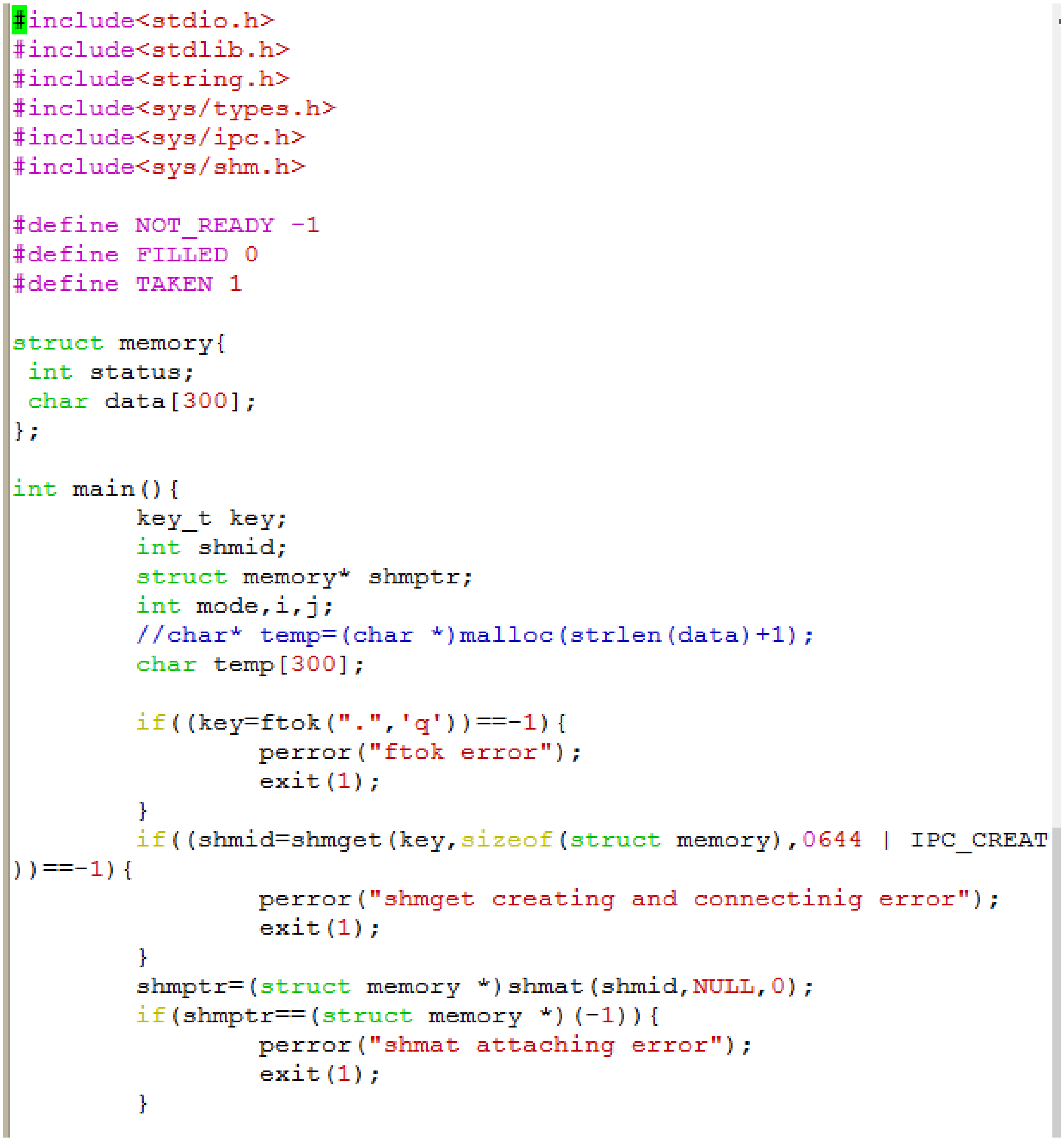
Parent: Ask a string from User and pass it to child process using shared memory.

Child: Read the string from shared memory and reverse it. Pass the reversed string to Parent process.

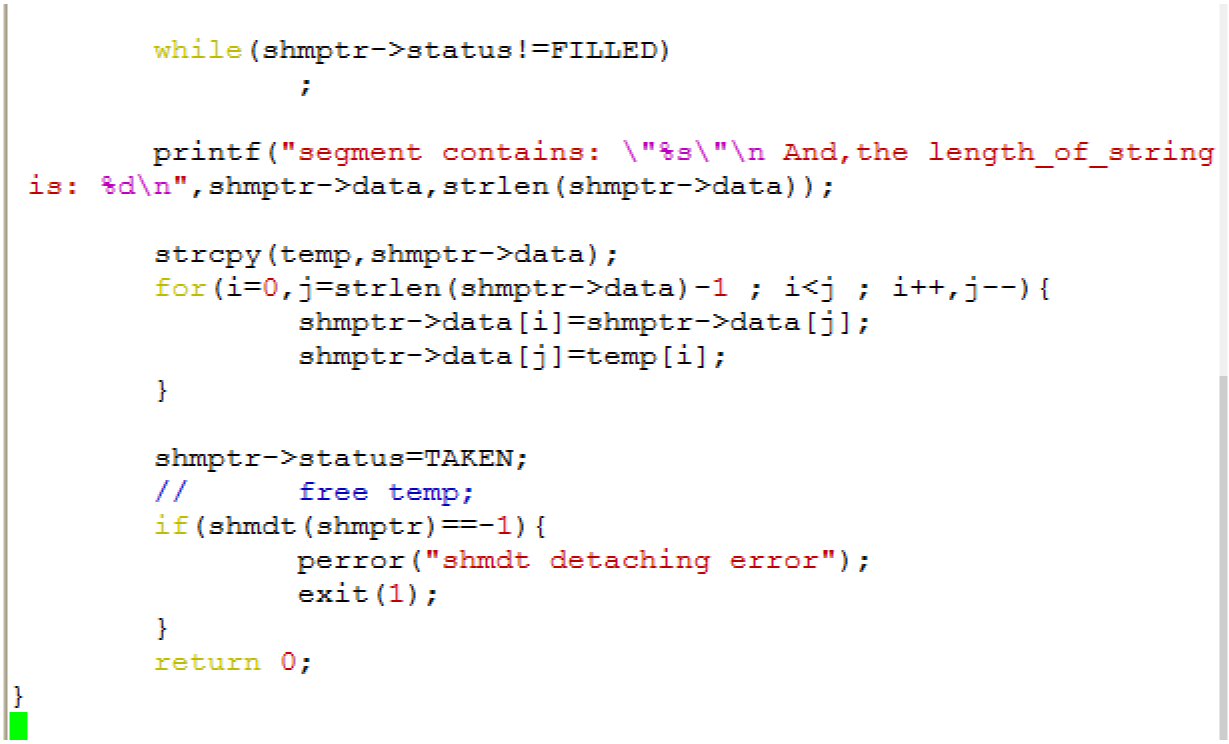
Parent: Read the string from shared memory and print it on screen



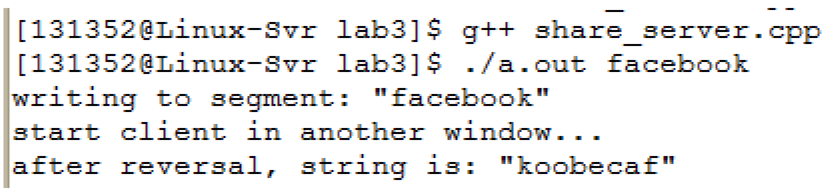
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/ipc.h>
#include<sys/shm.h>

#define NOT_READY -1
#define FILLED 0
#define TAKEN 1

struct memory{
 int status;
 char data[300];
};

int main(int argc,char* argv[]){
        key_t key;
        int shmid;
        struct memory* shmptr;

        //      shmptr->data=(char *)malloc(sizeof(struct memory));
        //       shmptr->data=(char *)malloc(300);

        if(argc>2){
                fprintf(stderr,"more arguments \n");
                exit(1);
        }
        if((key=ftok(".",'q'))==-1){
                perror("ftok error");
                exit(1);
        }
        if((shmid=shmget(key,sizeof(struct memory),0644 | IPC_CREAT))==-1){
                perror("shmget creating and connectig error");
                exit(1);
        }
        shmptr=(struct memory *)shmat(shmid,NULL,0);
        if(shmptr==(struct memory *)(-1)){
                perror("shmat attaching error");
                exit(1);
        }
        if(argc==2){
                printf("writing to segment: \"%s\"\n",argv[1]);
                shmptr->status=NOT_READY;
                strncpy(shmptr->data,argv[1],sizeof(struct memory));
        }
        printf("start client in another window... \n");
        shmptr->status=FILLED;

        while(shmptr->status!=TAKEN)
                sleep(1);
        printf("after reversal, string is: \"%s\"\n",shmptr->data);

        if(shmdt(shmptr)==-1){
                perror("shmdt detaching error");
                exit(1);
        }
        shmctl(shmid,IPC_RMID,NULL);

        return 0;
}
```



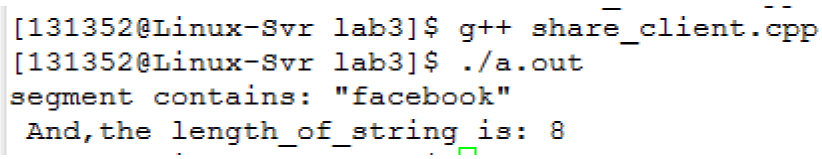
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<sys/ipc.h>
#include<sys/shm.h>

#define NOT_READY -1
#define FILLED 0
#define TAKEN 1

struct memory{
 int status;
 char data[300];
};

int main(){
        key_t key;
        int shmid;
        struct memory* shmptr;
        int mode,i,j;
        //char* temp=(char *)malloc(strlen(data)+1);
        char temp[300];

        if((key=ftok(".",'q'))==-1){
                perror("ftok error");
                exit(1);
        }
        if((shmid=shmget(key,sizeof(struct memory),0644 | IPC_CREAT
))==-1){
                perror("shmget creating and connectinig error");
                exit(1);
        }
        shmptr=(struct memory *)shmat(shmid,NULL,0);
        if(shmptr==(struct memory *)(-1)){
                perror("shmat attaching error");
                exit(1);
        }
```



```
        while(shmptr->status!=FILLED)
                ;

        printf("segment contains: \"%s\"\n And,the length_of_string
 is: %d\n",shmptr->data,strlen(shmptr->data));

        strcpy(temp,shmptr->data);
        for(i=0,j=strlen(shmptr->data)-1 ; i<j ; i++,j--){
                shmptr->data[i]=shmptr->data[j];
                shmptr->data[j]=temp[i];
        }

        shmptr->status=TAKEN;
        //      free temp;
        if(shmdt(shmptr)==-1){
                perror("shmdt detaching error");
                exit(1);
        }
        return 0;
}
```



```
[131352@Linux-Svr lab3]$ g++ share_server.cpp
[131352@Linux-Svr lab3]$ ./a.out facebook
writing to segment: "facebook"
start client in another window...
after reversal, string is: "koobecaf"
```



```
[131352@Linux-Svr lab3]$ g++ share_client.cpp
[131352@Linux-Svr lab3]$ ./a.out
segment contains: "facebook"
 And,the length_of_string is: 8
```