

```

#define CLE 312

struct message{
    long type;
    char numerocarte[16];
    bool valid;
}

main() {
    struct message mess;

    int pid;
    int fileid;
    int tubenommeid;

    char numcarte[16];
    char date[4];
    char pidaff[10];
    mode_t mode;
    mode = S_IRUSR | S_IWUSR;

    pid = fork();
    if (pid == 0){
        fileid = msgget((key_t)CLE, 0700|IPC_CREAT);
        while(true){
            msggrsv((fileid, &mess, sizeof(struct message)-
4, getpid(), 0);
            if(mess.valid)
                printf("la carte %s est valide",
mess.numerocarte);
            }else{
                printf("la carte %s n'est pas valide",
mess.numerocarte);
            }
        }
    }else {
        mkfifo ("testcarte", mode);
        tubenommeid = open ("testcarte", O_WRONLY);
        while(true){
            getCarteInformation(&numcarte, &date);
            scanf(pidaff, %d", pid);
            write (tubenommeid, numcarte, 16);
            write (tubenommeid, date, 4);
            write (tubenommeid, pidaff, 10);
        }
    }
}

```

decodeur.c

```

#define CLE 312

```

```

main() {
    struct message mess;

    int pid1, pid2;
    int pip1[2], pip2[2];
    int fileid;
    int tubenommeid;

    char numcarte[16];
    char date[4];
    char pidaff[10];

    bool fils1;
    fils1 = 1;

    pipe(pip1);
    pipe(pip2);

    pid1 = fork();
    if (pid1 == 0){
        //dans le fils 1
        close(pip1[0]);
        fileid = msgget((key_t)CLE, 0700|IPC_CREAT);
        while(true) {
            read (pip1[1], numcarte, 16);
            read (pip1[1], date, 4);
            read (pip1[1], pidaff, 10);
            strcpy(mess.numerocarte,numcarte);
            mess.type = atoi(pidaff);
            mess.valid = testCarte(numcarte, date);
            msgsnd(fileid, &mess, sizeof(struct message)-4,
0);
        }
    }else {
        //dans le pere
        pid2 = fork();
        if (pid2 == 0){
            //dans le fils 2
            close(pip2[0]);
            fileid = msgget((key_t)CLE, 0700|IPC_CREAT);
            while(true) {
                read (pip2[1], numcarte, 16);
                read (pip2[1], date, 4);
                read (pip2[1], pidaff, 10);
                mess.numerocarte = atof(numcarte);
                mess.type = atoi(pidaff);
                mess.valid = testCarte(numcarte, date);
                msgsnd(fileid, &mess, sizeof(struct
message)-4, 0);
            }
        }
    }
}

```

```
}else {
    // dans le pere
    tubenommeid = open ("testcarte", O_RDONLY);
    close(pip2[1]);
    close(pip2[1]);
    while (true) {
        read (tubenommeid, numcarte, 16);
        read (tubenommeid, date, 4);
        read (tubenommeid, pidaff, 10);
        if(filsl){
            filsl = 0;
            write (pip1[0], numcarte, 16);
            write (pip1[0], date, 4);
            write (pip1[0], pidaff, 10);
        }else{
            write (pip2[0], numcarte, 16);
            write (pip2[0], date, 4);
            write (pip2[0], pidaff, 10);
            filsl = 1;
        }
    }
}
}
}
```