

```

#include<stdio.h>

typedef struct{
    double reel;
    double imaginaire;
}complexe;

void charger_complexe(complexe *p_complexe, double reel, double imaginaire↵
)
{
    p_complexe->reel=reel;
    p_complexe->imaginaire=imaginaire;
}

void recuperer_complexe(complexe *p_complexe, double *p_reel, double *p_i↵
maginaire)
{
    *p_reel=p_complexe->reel;
    *p_imaginaire=p_complexe->imaginaire;
}

void ajouter_complexe(complexe *p_somme, complexe *p_complexe1, complexe ↵
*p_complexe2)
{
    p_somme->reel=p_complexe1->reel + p_complexe2->reel;
    p_somme->imaginaire=p_complexe1->imaginaire+ p_complexe2->imaginaire;
}

void multiplier_complexe(complexe *p_produit, complexe *p_complexe1, comp↵
lexe *p_complexe2)
{
    p_produit->reel=p_complexe1->reel*p_complexe2->reel - p_complexe1->imag↵
inaire*p_complexe2->imaginaire;
    p_produit->imaginaire=p_complexe1->reel*p_complexe2->imaginaire + p_com↵
plexe1->imaginaire*p_complexe2->reel;
}

void afficher_complexe(complexe *p_complexe)

{ printf(" %g +i %g /n ",p_complexe->reel, p_complexe->imaginaire);}

int main()
{
    complexe C1,C2;

    charger_complexe(&C1,10.,5.);
    printf(" %g +i %g /n",C1.reel, C1.imaginaire);
    afficher_complexe(&C1);
    charger_complexe(&C2,5.,10.);
    afficher_complexe(&C2);
    ajouter_complexe(&C1,&C2,&C1);
    afficher_complexe(&C1);
    multiplier_complexe(&C1,&C2,&C2);
    afficher_complexe(&C1);
    return 0;
}

```