

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>

#define M_PI 3.14159265358979

struct cart{
    double re;
    double im;
};

struct polar{
    double mo;
    double fa;
};

double gAr(double gr){
    return(M_PI*gr/180.0);
}

double rAg(double rd){
    return(180.0*rd/M_PI);
}

int firstMenu(){
    char r;

    do{
        printf("\n\n\t1) Introduir elements");
        printf("\n\n\tS) Sortir\n\n");
        r = getch();
    }while(!(r == '1') || (r == 's') || (r == 'S')));

    if(r == '1')
        return 1;
    return 0;
}

void showPolar(struct polar a, char c){
    printf("\n\n %c = %lf|_ %lf ",c,a.mo,rAg(a.fa));
}

void showCart(struct cart a){
    printf("= (%lf + i%lf)\n",a.re,a.im);
}

void cAp(struct cart c, struct polar *p){
    p->mo = sqrt(c.re * c.re + c.im * c.im);
    p->fa = atan(c.im/c.re);
}

void pAc(struct polar p,struct cart *c){
    c->re = p.mo * cos(p.fa);
    c->im = p.mo * sin(p.fa);
}

int cartChosen(char elem){
    char r;

    do{
        printf("\n\nRepresentar %c en cartesianes (c) o polars (p)? . Sortir (s). (C/P/S)? : ",elem);
        r = getch();
    }while(!(r == 'c') || (r == 'C') || (r == 'p') || (r == 'P') || (r == 's') || (r == 'S')));

    if((r == 'c') || (r == 'C'))
        return 1;
    if((r == 's') || (r == 'S'))
        exit(0);
    return 0;
}
```

```

void askElement(struct polar *p, struct cart *c, char el){
    if(cartChosen(el)){
        printf("\n\n\n\nPart real de %c : ",el);
        scanf("%lf",&c->re);
        printf("\n\n\n\nPart imaginària de %c : ",el);
        scanf("%lf",&c->im);
        cAp(*c,p);
    }else{
        printf("\n\n\n\nMòdul de %c : ",el);
        scanf("%lf",&p->mo);
        printf("\n\n\n\nFase de %c : ",el);
        scanf("%lf",&p->fa);
        p->fa = gAr(p->fa);
        pAc(*p,c);
    }
}

int mainMenu(struct polar *xP, struct cart *xC,
            struct polar *yP, struct cart *yC,
            char x, char y){
    int whichOne;

    do{
        showPolar(*xP,x);
        showCart(*xC);
        showPolar(*yP,y);
        showCart(*yC);
        printf("\n\n\nn1) Suma\n");
        printf("2) Resta\n");
        printf("3) Multiplicació\n");
        printf("4) Divisió\n");
        printf("5) Canviar l'element %c\n",x);
        printf("6) Canviar l'element %c\n",y);
        printf("7) Sortir\n\n\n");
        whichOne = getch();
        //printf("%c",whichOne);
    }while (!((whichOne >= '1') && (whichOne <= '7')));

    if (whichOne == '7') whichOne = 0;
    return whichOne;
}

void suma(struct cart aC, struct cart bC,
          struct polar *rP, struct cart *rC){
    rC->re = aC.re + bC.re;
    rC->im = aC.im + bC.im;
    cAp(*rC,rP);
}

void resta(struct cart aC, struct cart bC,
           struct polar *rP, struct cart *rC){
    rC->re = aC.re - bC.re;
    rC->im = aC.im - bC.im;
    cAp(*rC,rP);
}

void mult(struct polar aP, struct polar bP,
          struct polar *rP, struct cart *rC){
    rP->mo = aP.mo * bP.mo;
    rP->fa = aP.fa + bP.fa;
    pAc(*rP,rC);
}

int div(struct polar aP, struct polar bP,
        struct polar *rP, struct cart *rC){
    if(bP.mo){
        rP->mo = aP.mo / bP.mo;
        rP->fa = aP.fa - bP.fa;
        pAc(*rP,rC);
        return 1;
    }
    return 0;
}

void main(){
    struct cart aC, bC, rC;
    struct polar aP, bP, rP;
    int sel;
}

```

```

if(firstMenu()){
    askElement(&aP,&aC,'a');
    askElement(&bP,&bC,'b');
    while(sel = mainMenu(&aP,&aC,&bP,&bC,'a','b')){
        switch(sel){
            case '1':
                suma(aC,bC,&rP,&rC);
                showPolar(rP,'r');
                showCart(rC);
                break;
            case '2':
                resta(aC,bC,&rP,&rC);
                showPolar(rP,'r');
                showCart(rC);
                break;
            case '3':
                mult(aP,bP,&rP,&rC);
                showPolar(rP,'r');
                showCart(rC);
                break;
            case '4':
                if(div(aP,bP,&rP,&rC)){
                    showPolar(rP,'r');
                    showCart(rC);
                }else
                    printf("\nNo es pot dividir per zero !!!!\n");
                break;
            case '5':
                askElement(&aP,&aC,'a');
                break;
            case '6':
                askElement(&bP,&bC,'b');
                break;
        }
    }
}
}

```