**POINTERI LA FUNCTII**

**Declarare :**

Tip\_returnat\_de\_functie (\*variabila\_pointer)(tipurile\_parametrilor\_functiei);

Int suma(int a,int b){

Return a+b;

}

Int produs(int a ,int b){

Return a\*b;

}

Int (\*pf)(int,int); /\*pf e variabila de tip pointer la functie, unde functia are sablonul intoarce un intreg si are parametrii catre intregi\*/

Pf=&suma;

S=suma(1,3); ->apelare directa

S=(\*pf)(1,3); ->apelare indirect ⬄s=pf(1,3);

OBS! Numele unei functii este pointer catre functia respectiva.

Suma ⬄&suma

Ex: s=1+2+…+n

Int suma1 (int n){

Int i,s=0;

For(i=1;i<=n;i++)

S1+=i;

Return s;

}

S2=1^2+2^2+..+n^2

Int suma2(int n){

....

...

S+=i\*i;

}

S3=[sin(1)]+[sin(2)]+...+[sin(n)]

Int suma3(int n){

...

...

S+=(int)sin(i);

...

}

**Pe caz general:**

S=f(1)+f(2)+..+f(n);

f- arbitrara de forma int f(int x)

ex: int suma(int n,int (\*pf)(int)){

int i,s;

S=0;

For(i=1;i<=n;i++)

S+=(\*pf)(i); // s+=pf(i);

Return s;

}

Int f1(int x) //f1 e termenul general al sumei s1

{return x;}

Int f2(int x){

Return x\*x;

}

Int f3(int x){

Return(int)sin(x);

}

S1=suma(n,&f1); // s1=suma(n,f1)

S2=suma(n,&f2);

S3=suma(n,&f3):

**TABLOURI DE POINTERI LA FUNCTII**

**Declarare:**

Tip\_returnat\_de\_functie(\*nume\_tablouri[dim])(tipurile\_parametrilor)

Double if(double x){

Return 2\*x;

}

Double(\*tpf[3])(double) //tablou cu 3 elemente care intorc valorile functiei de tip double

Tpf[0]=&f; //tpf[0]=f;

Tpf[1]=&sin; //tpf[1]=sin; functia sin apartine <math.h>

...

Y=tpf[o](1);