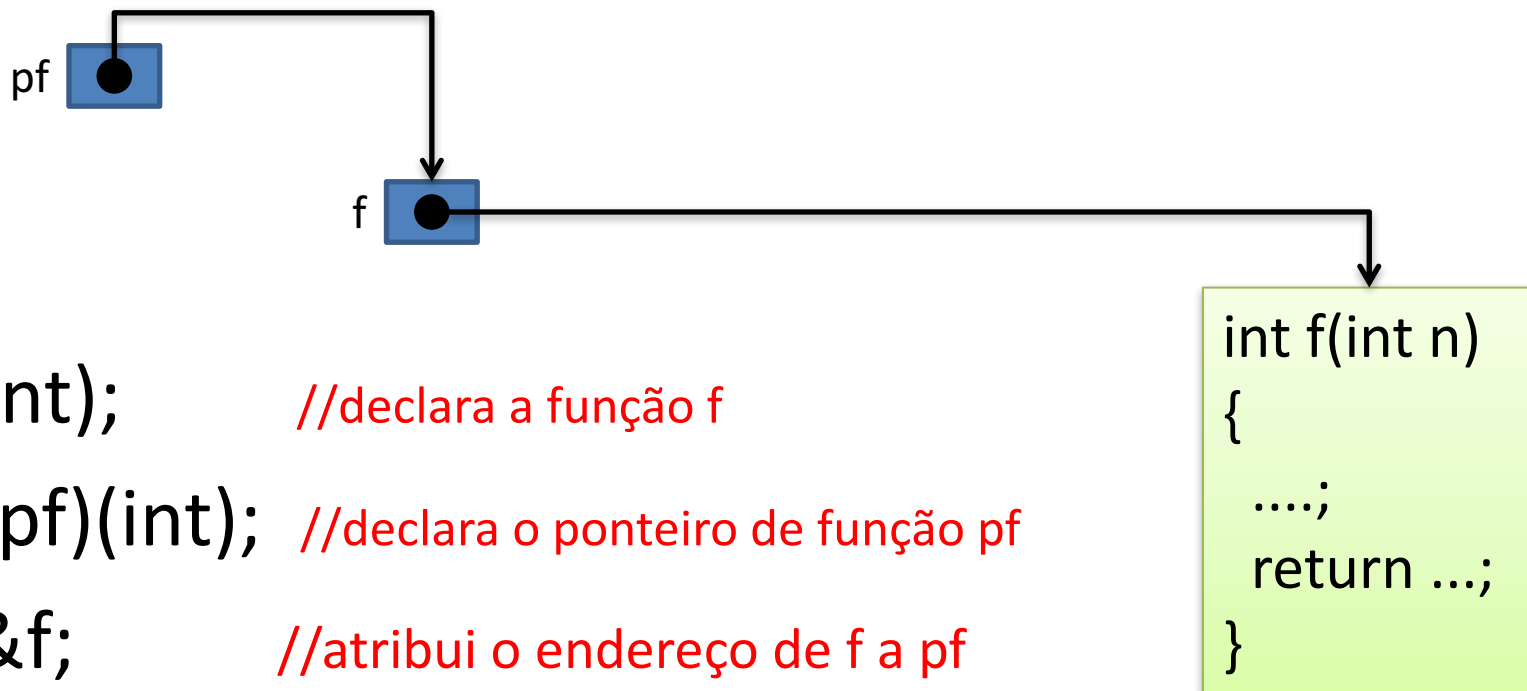


Estrutura de Dados e Algoritmos

Ponteiros de Funções

- As linguagens C/C++ permitem o acesso ao endereço de memória de funções → Ponteiros para Funções;
- Declaração de um ponteiro para função:
TipoRetorno (*nomeponteiro)(<lista de parâmetros>);
- Um ponteiro para função é definido pelo tipo de retorno e por sua lista de parâmetros:

- Em C/C++, o nome de uma função é um ponteiro.



- O ponteiro para função pode ser usado para fazer uma chamada a função, com a mesma sintaxe de chamadas comuns.

```
void * (*ptr) (size t, size t);  
ptr = calloc;  
...  
int *p = (int *) ptr(2, 8);
```

- Ponteiros para função permitem a construção de programas dinâmicos e podem aumentar a organização do código.

```
#include<stdio.h>
#include "exemplo.c"
main()
{
    float R, R2;
    float (*p) (int i, int j);
    p = exemplo;
    R  = p(12,45);
    R2 = (*p)(12,45);
    printf("R= %f, R2 = %f", R, R2);
}
```

```
float exemplo(int n1, int n2)
{
    return 3.14159*n1*n2;
}
```

- Parâmetros do tipo ponteiro de função.

```
#include<stdio.h>
#include "funcoes.c"
main()
{
    float psq, psc;
    psq = AplicaF(5,quadrado);
    psc = AplicaF(5,cubo);
    printf("Soma dos quadrados: %f",psq);
    printf("Soma dos cubos: %f",psc);
}
```

```
float quadrado(int pnum)
{
    return pnum*pnum;
}
float cubo(int pnum)
{
    return pnum*pnum*pnum;
}

float AplicaF(int n, float (*f) (int k))
{
    int ptermo;
    float psoma = 0;
    for(ptermo=1; ptermo<=n; ptermo++)
        psoma += f(ptermo);
    return psoma;
}
```

- Vetores de ponteiros para funções permite a implementação de um conjunto de diferentes funções que podem ser selecionadas por índices.
- Sintaxe geral:

TipoRetorno (*PtrFunc[Qtde])(<lista de parâmetros>);

- Exemplos:

```
void (*series[10]) (unsigned int pq);
```

```
double (*calculos[3]) (double *pvalores, unsigned int n);
```

```
char* (*formato[5]) (char *frase, unsigned int n);
```


- `func` → nome de um objeto;
- `func[]` → é um vetor;
- `(*func[])` → é um vetor de ponteiros;
- `(*func[])()` → é um vetor de ponteiros para funções;
- `int (*func[])()` → é um vetor de ponteiros para funções;

- Pode-se atribuir o endereço de funções já existentes ao vetor:

```
int func1(int i, int j);  
int func2(int i, int j);  
int (*func[])(int, int) = {func1, func2};
```