

---

# Laboratório de Introdução à Ciência da Computação I

## **Aula 11 – Ponteiros**

Professor:  
Jó Ueyama

# Declaração de ponteiros

---

Uso do operador de indireção -> \*

tipo\_variável \*nome\_ponteiro

int \*p;

char \*q;

float \*r,\*s;

# Inicialização de ponteiros

---

```
int *p; //declaração do ponteiro p;
```

```
int x = 2; //declaração de uma variável x;
```

```
//inicialização do ponteiro p (aponta para a variável x)
```

```
p = &x; //( & ) operador para obter o endereço da variável
```

# Utilizando ponteiros

---

Dado:

```
int *p, x = 4;
```

```
p = &x;
```

Qual a diferença?

```
printf("%d %d", p, *p);
```

Qual o resultado?

```
printf("%d %d \n", p+1, *p+1);
```

# Ponteiros para vetores

---

1 - int vet[100], \*p\_vet;

2 - p\_vet = vet;

3 - p\_vet = &vet[0];

Qual a diferença entre as linhas 2 e 3?

# Incrementando/decremento de ponteiros

---

```
int vet[100], *p_vet;
```

```
p_vet = vet;
```

```
p_vet++; // aponta para o próximo elemento do vetor
```

```
p_vet--; // aponta para o elemento anterior do vetor
```

```
p_vet += 4; //aponta para a posição atual do vetor + 4;
```

*Dado um ponteiro que aponta para um endereço 100, se o mesmo por incrementado de 1, para qual endereço o mesmo apontará?*

# Incrementando/decremento de ponteiros

---

```
int vet[] = {10,11,12,13}, *p_vet, cnt;
p_vet = vet;
for(cnt=0; cnt<3; cnt++){
    printf("\n %d", *p_vet++);
}
```

```
p_vet = vet;
for(cnt=0; cnt<3; cnt++){
    printf("\n %d", *++p_vet);
}
```

Qual a diferença do resultado impresso pelos dois “printf”?

# Ponteiros com membros de estruturas

---

```
struct dados{  
    int *valor;  
    int *taxa;  
}primeira;
```

```
int custo = 10, juros 5;
```

```
primeira.valor = &custo;  
primeira.taxa = &juros;
```



# Exercícios

---

- Exercícios 3 e 4 da Lista (Prof. Vanderlei)