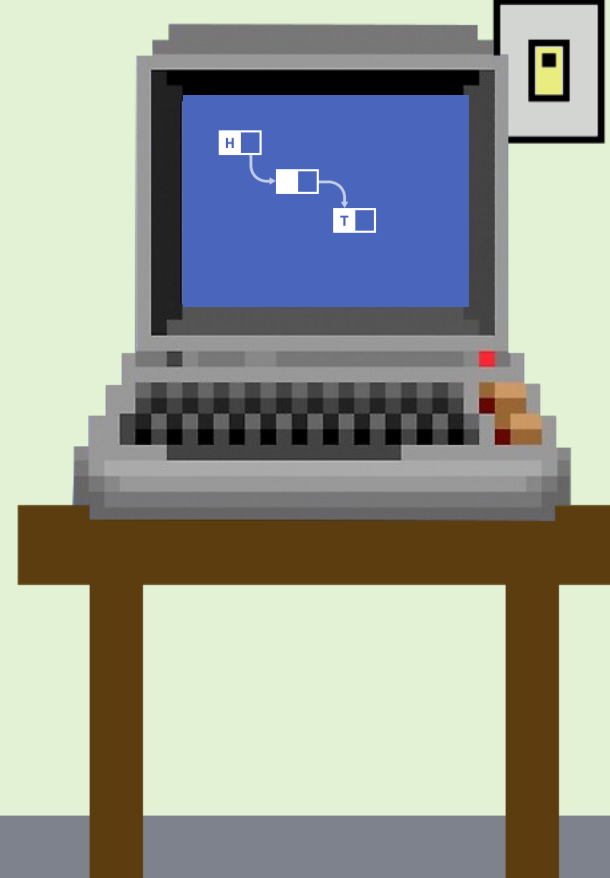
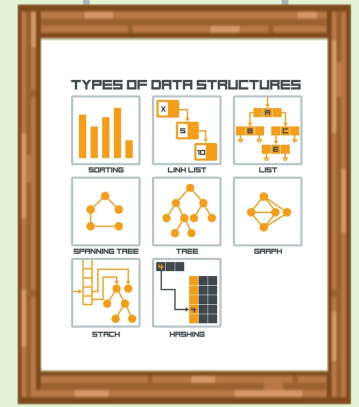


Registros

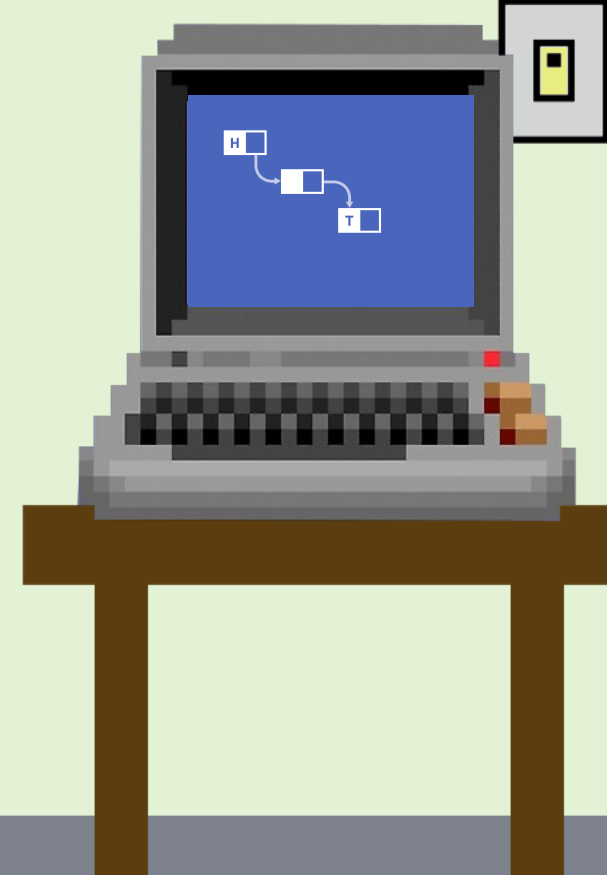
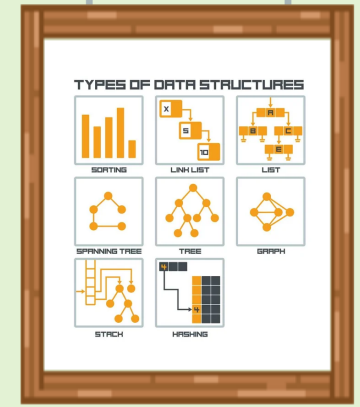
Estruturas de Dados I



Registros

- Estrutura de dados heterogênea
 - Conjunto de variáveis (podem ter tipos diferentes)
- Cada variável é um campo do registro (field)
- Em C são conhecidos como structs
 - Representam estruturas de dados mais complexas
- Exemplo
 - Data (dd/mm/aaaa)

```
struct {  
    int dia;  
    int mes;  
    int ano;  
} data;
```



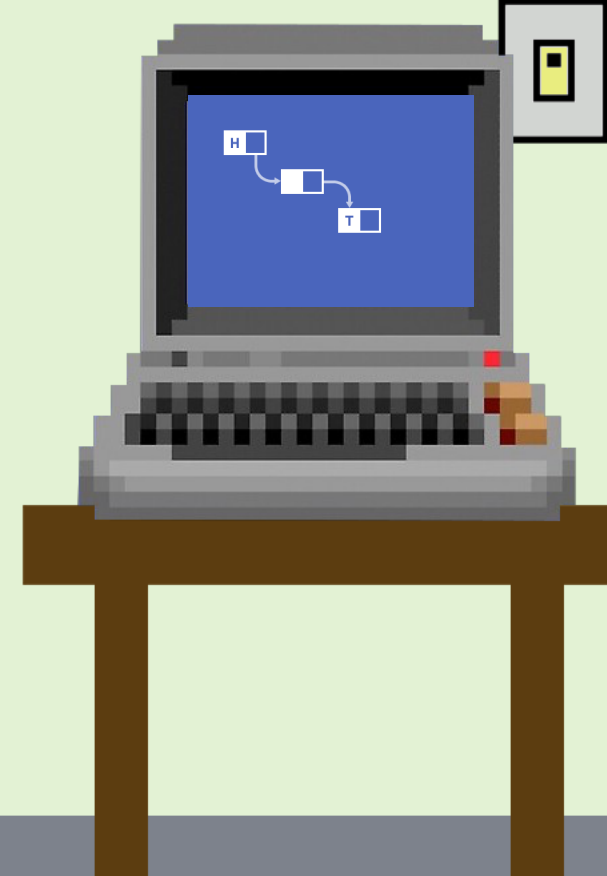
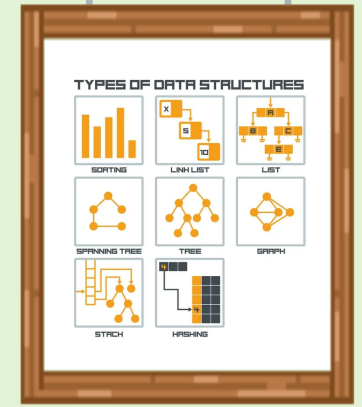
Registros

- Pode definir um novo tipo de dado (typedef)

```
typedef struct {  
    int dia;  
    int mes;  
    int ano;  
} data;
```

- Acesso aos campos utilizando . (ponto)

```
data nasc_cristo;  
nasc_cristo.dia=25;  
nasc_cristo.mes=12;  
nasc_cristo.ano=0;
```



Registros e Ponteiros

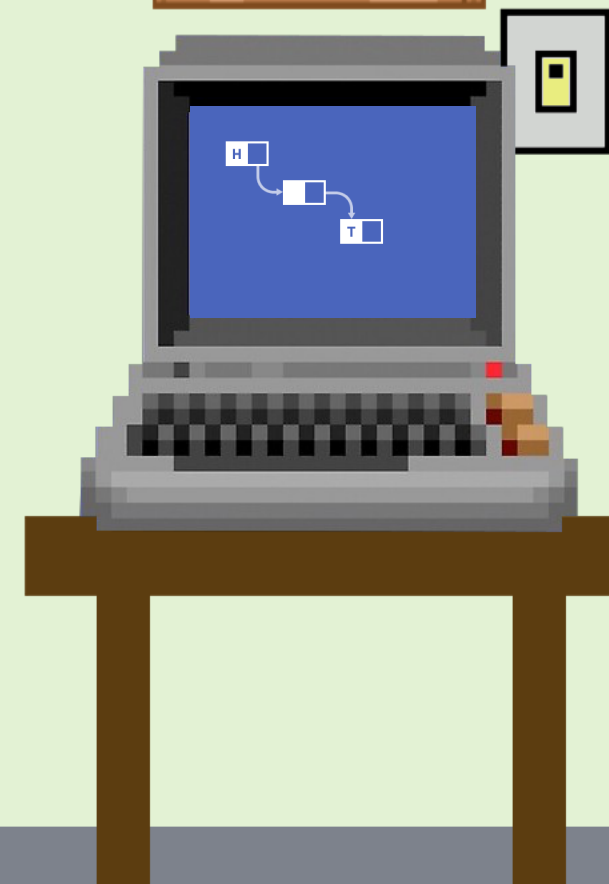
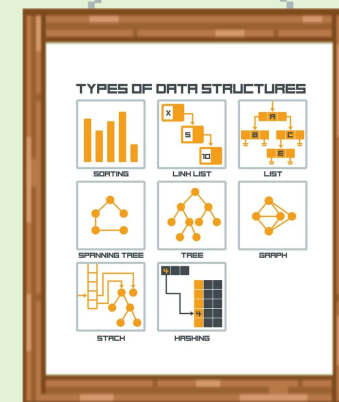
- Podemos criar ponteiros para registros

```
data nasc_cristo;  
data *ptr_data = &nasc_cristo;
```
- Acesso aos campos

```
(*ptr_data).dia = 25; // notação confusa  
*ptr_data.dia = 25; // errado
```
- Operador (->)

```
ptr_data->dia = 25;
```
- Exemplo

```
printf("%d", ptr_data->dia);
```



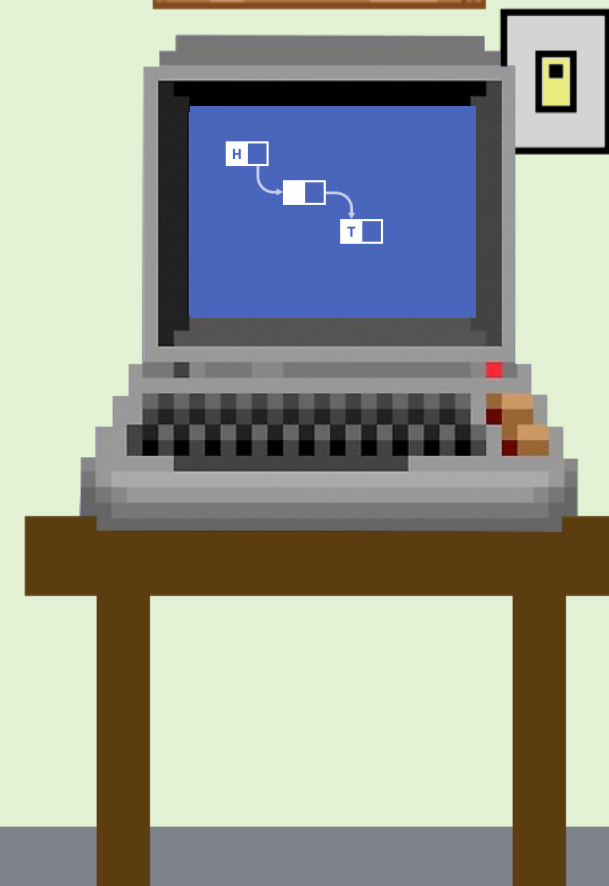
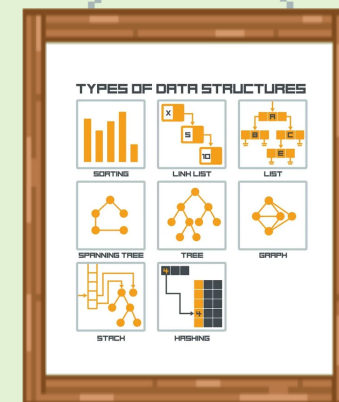
Registros e Ponteiros

- Registros podem conter campos do tipo ponteiro

```
typedef struct {  
    char *nome;  
    int telefone;  
    data *nascimento;  
} contato;
```

- Acesso aos campos

```
char n[100];  
data nasc;  
contato amigo;  
amigo.nome = n;  
amigo.nascimento = &nasc;
```



Alocação Dinâmica

- Podemos alocar registros dinamicamente na memória

- Função malloc

- Função free

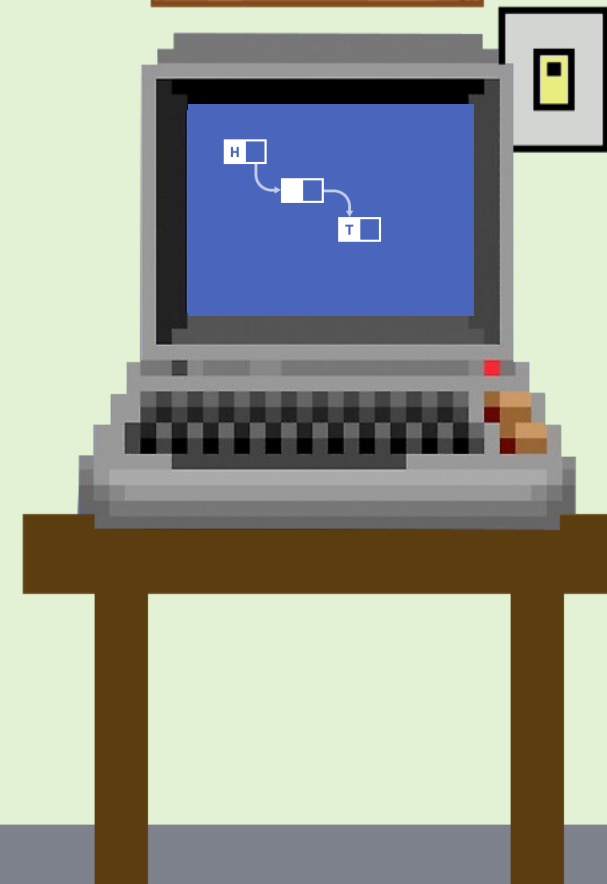
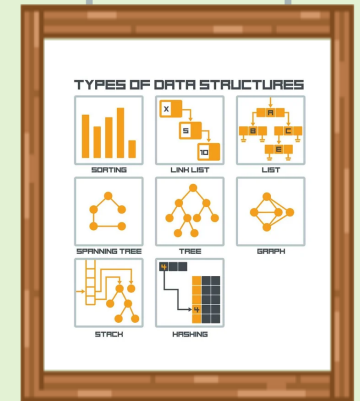
- Exemplo

```
char n[100];
```

```
contato *cnt = (contato *)malloc(sizeof(contato));
```

```
cnt->nascimento = (data *)malloc(sizeof(data));
```

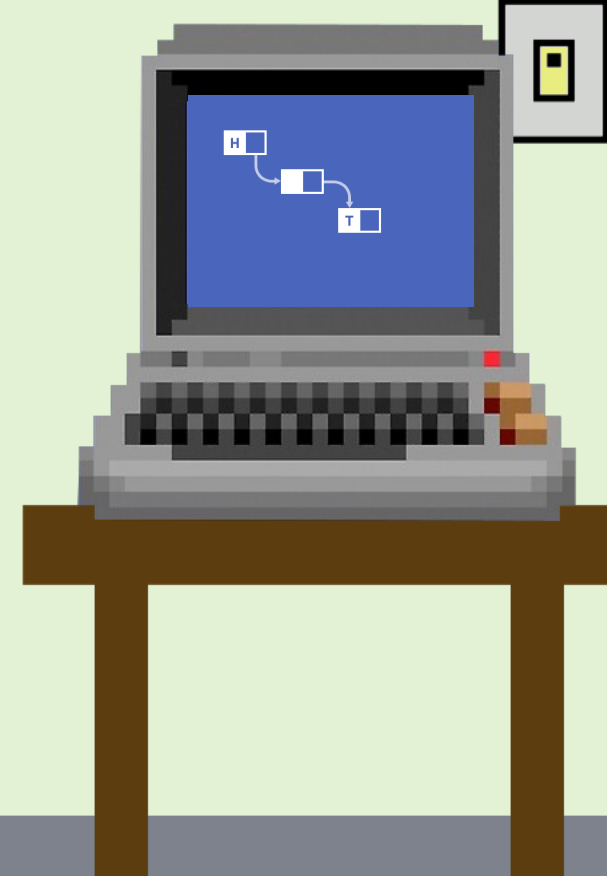
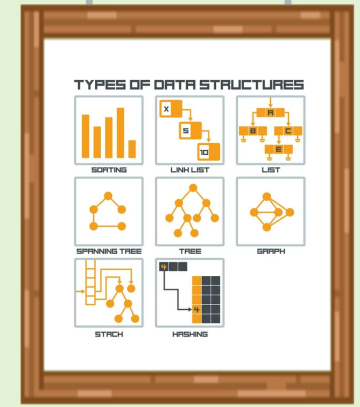
```
cnt->nome = n;
```



Alocação Dinâmica

- A função malloc também aloca vetores de registros
- Exemplo

```
data *datas = (data *)malloc(sizeof(data) * total);  
scanf("%d", &datas[0].dia);  
scanf("%d", &datas[0].mes);  
scanf("%d", &datas[0].ano);  
  
printf("%02d/%02d/%04d\n", datas[0].dia,  
      datas[0].mes, datas[0].ano);
```



Atividade

- Implemente uma agenda de contatos que solicita o tamanho do vetor a ser alocado dinamicamente
 - Dados de cada contato
 - Nome
 - Endereço
 - Rua - char[50]
 - Casa - int
 - Cidade - char[30]
 - Telefone - int
 - Data de Nascimento
 - Dia
 - Mês
 - Ano

