

Exercício - 01 - Mateus Filipe De Lima Souza

Declaração de variáveis

- **int -**

Armazena variáveis do tipo inteira ignorando algo que venha depois da vírgula caso seja digitado um decimal. Armazena 4 Bytes;

- **float -**

Armazena variáveis do tipo flutuante (que seriam as variáveis que possuem casas após a vírgula exemplo: 1,5) Pode ser utilizada para armazenar variáveis inteiras, mas ainda assim serão do tipo flutuante, caso digite 1, reconhece normalmente. Armazena 4 Bytes;

- **double -**

Também do tipo flutuante, mas com maior capacidade de reconhecimento de casas após a vírgula maior ou igual ao reconhecimento de um variável do tipo float. Armazena 8 Bytes;

- **long -**

Capacidade de armazenamento maior do que a de int, ou igual a ele. Armazena 4 bytes;

- **long long -**

Capacidade de armazenamento maior ainda do que long chegando a armazenar 8 bytes;

Modificadores de tipos de dados

- **short** -

Vai fixar a faixa das variáveis declaradas para -32.768 até 32.768. Se você declarar `int` normalmente ele é padronizado como `long int`.

Exemplo: `short int n;`

- **long** -

Aumenta a faixa dos números inteiros de -2.147.483.648 até -2.147.483.647, mas lembrando que ao se usar `int` normalmente ela já vem como **long int**. Long também pode ser utilizado em float, mas passa a se tornar **double**.

- **Unsigned / signed** -

Ao definirmos um **int** normalmente ele vem como **unsigned**, pois ele não tem sinal definido, ao utilizar o **signed** fixa-se a faixa normal para um número inteiro.

Conversões: Implícita e explícita

- **Implícita:**

Já é realizada pelo próprio programa sem a necessidade de edição, exemplificando:

```
char letra = 'M';
```

```
int k;
```

```
int j;
```

```
k = letra; Aqui k recebe o valor do código da letra M;
```

```
j = k; Aqui j vai receber o código de k;
```

- **Explícita:**

Realizada na hora da programação no meio do código, exemplificando:

```
float m;
```

```
[...]
```

```
if(int(m)%2==0)
```

```
cout<<"Par"<<endl;
```