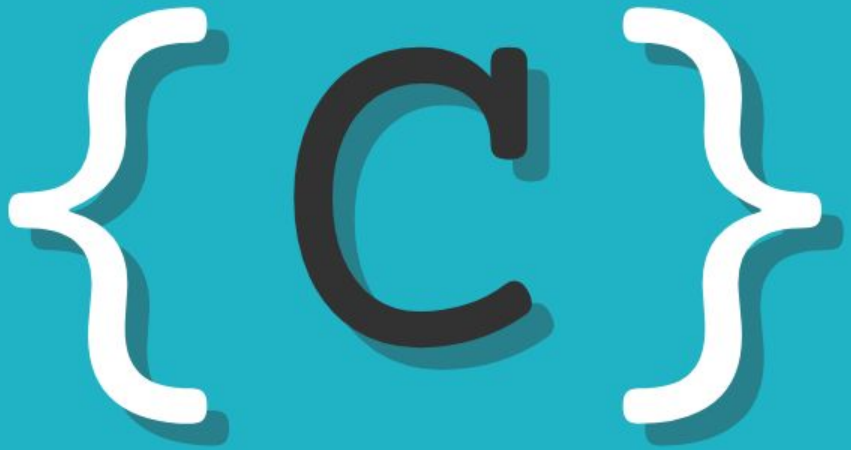
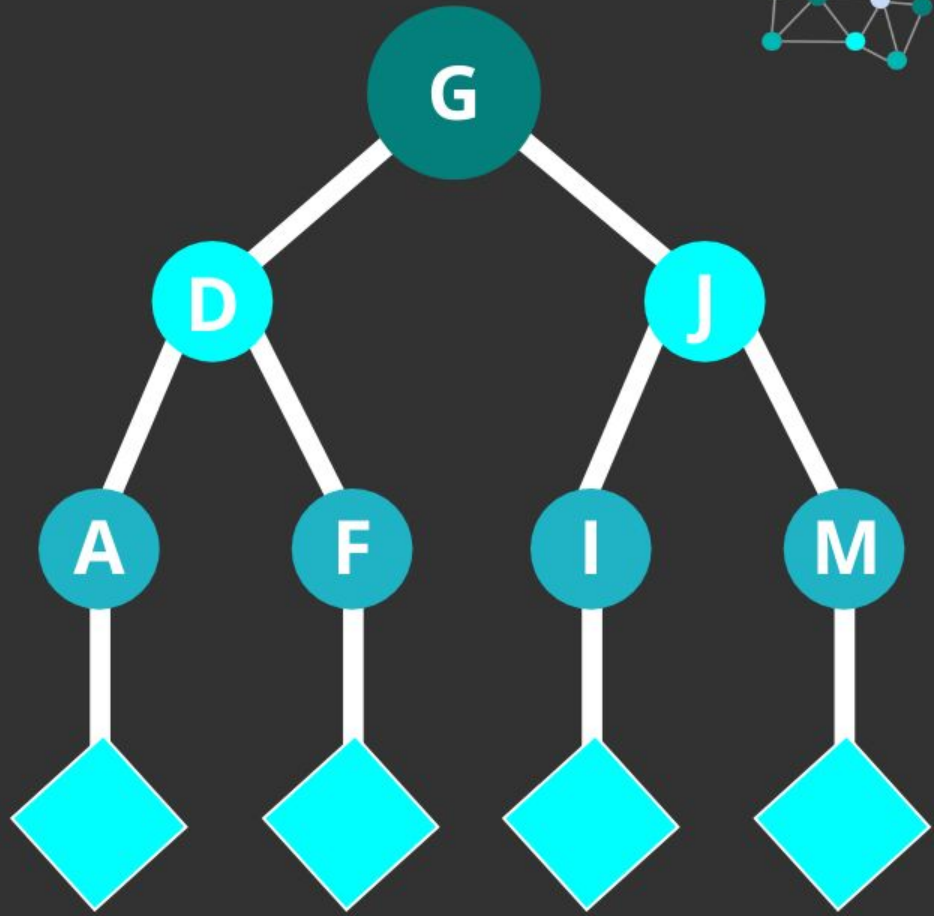
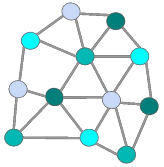
Leonardo de
Abreu
Schmidt



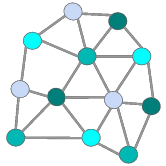


Recursão



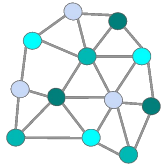
- A recursão é uma ferramenta pela qual podemos chamar recursivamente uma função.
- Ou seja, chamamos a função dentro dela mesma.

Recursão



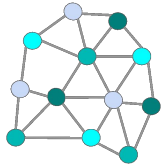
```
int main() {
    int v = potencia(2,4);
    printf("%d",v);
    return 0;
}
```

Recursão



```
int main() {
    int v = potencia(2,4);
    printf("%d",v);
    return 0;
}
```

Recursão

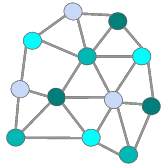


```
int v = potencia(2,4);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4

Recursão

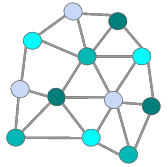


```
int v = potencia(2,4);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4

Recursão

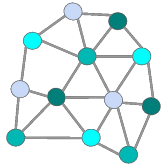


```
int v = potencia(2,4);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4

Recursão

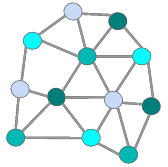


```
int v = 2 * potencia(2,3);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4

Recursão

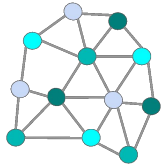


```
int v = 2 * potencia(2,3);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4
2	3

Recursão

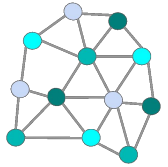


```
int v = 2 * potencia(2,3);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4
2	3

Recursão

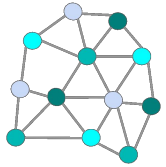


```
int v = 2 * potencia(2,3);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4
2	3

Recursão

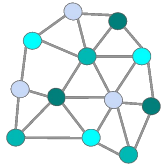


```
int v = 2 * 2 * potencia(2,2);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4
2	3

Recursão

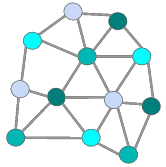


```
int v = 2 * 2 * potencia(2,2);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4
2	3
2	2

Recursão

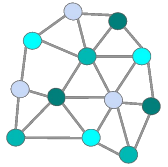


```
int v = 2 * 2 * potencia(2,2);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4
2	3
2	2

Recursão

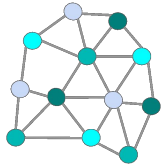


```
int v = 2 * 2 * potencia(2,2);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4
2	3
2	2

Recursão

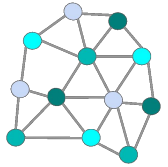


```
int v = 2 * 2 * 2 * potencia(2,1);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4
2	3
2	2

Recursão

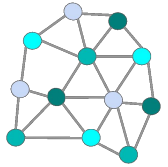


```
int v = 2 * 2 * 2 * potencia(2,1);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4
2	3
2	2
2	1

Recursão

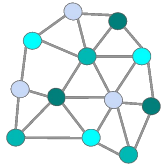


```
int v = 2 * 2 * 2 * potencia(2,1);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4
2	3
2	2
2	1

Recursão

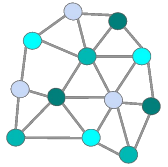


```
int v = 2 * 2 * 2 * potencia(2,1);
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

base	ep
2	4
2	3
2	2
2	1

Recursão

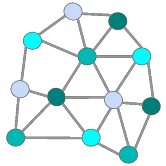


```
int v = 2 * 2 * 2 * 2;
```

```
int potencia(int base, int ep){
    if(ep==1){
        return base;
    }
    return base*potencia(base,ep-1);
}
```

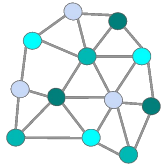
base	ep
2	4
2	3
2	2
2	1

Recursão



```c
int main() {
    int v = 2 * 2 * 2 * 2;
    printf("%d",v);
    return 0;
}
```

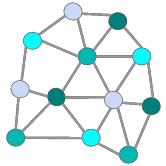
Recursão



```c
int main() {
    int v = 2 * 2 * 2 * 2;
    printf("%d",v);
    return 0;
}
```

Saída
16

Recursão



```c
int main() {
    int v = 2 * 2 * 2 * 2;
    printf("%d",v);
    return 0;
}
```

Leonardo de
Abreu
Schmidt