Recursão e algoritmos recursivos

– Propriedade: cada instância do problema contém uma instância menor do mesmo problema.

– Dizemos que tais problemas possuem estrutura recursiva.
– Para resolver uma instância de um problema desse tipo, podemos aplicar o seguinte método:

- Se a instância em questão for pequena, resolva-a diretamente.
- Senão:
  - reduza-a a uma instância menor do mesmo problema, aplique o método à uma instância menor, volte à instância menor, volte à instância original.

– O programador só precisa mostrar como obter uma solução da instância original a partir de uma solução da instância menor.
– A aplicação desse método produz um algoritmo recursivo.

Exemplo: Determina o valor de um elemento máximo de um vetor v[0, …, n-1].
Caso base: quando o vetor possui somente um elemento.
Caso recursivo:

máximo(v[0...n-1])
//função anexada no outro arquivo “.c”

Endereços e ponteiros

– Endereços:
A memória RAM (= Random Acess Memory) de qualquer computador é uma sequência é o endereço (= address) do byte.
Se x é o endereço de um byte então x+1 é o enderaço do byte seguinte
Cada variável de um programa ocupa um certo número de bytes consecutivos na memória do computador.
Ex.:
char – 1 byte
int – 4 bytes
double – 8 bytes

– O número exato de bytes de uma variável é dado pelo operador sizeof.
– O endereço de uma variável é dado pelo operador &.
– Assim, se *i* é uma variável então &i é seu endereço.
Exemplo: O segundo argumento da função scanf é o endereço da variável que deve receber o valor lido do teclado.

```
Int i;
scanf(“%d”, &i);
```

Ponteiros:
– um ponteiro (=apontador =operador) é um tipo especial de variável que armazena um endereço.
– um ponteiro pode ter valor NULL que é um endereço “inválido”.

- Se um ponteiro p armazena o endereço de uma variável *i*, podemos dizer “p aponta para i” ou “p é o endereço de i”.
– Se um ponteiro é diferente de NULL então *p é o valor da variável apontada por p.
Por exemplo, se *i* é uma variável e p=&i, então dizer “*p” é o mesmo que dizer “i”.
Tipos de ponteiros:
– Há vários tipos de ponteiros:
- ➔ ponteiros para bytes
- ➔ ponteiros para inteiros
- ➔ ponteiros para ponteiros para inteiros
- ➔ ponteiros para registros
- ➔ etc.

Para declarar um ponteiro p para um inteiro, escrevemos:
int *p;

função que calcule o maior e o menor entre 10 elementos de um vetor