Inspeção de Lista
Professor Wladimir

Problema soma3

Para encontrar a soma dos valores de três variáveis inteiras a, b e c, construímos o seguinte programa:

```
1  #include <stdio.h>
2
3  int a = 4, b = 6, c = 8;
4  int soma;
5
6  int main(){
7      soma = 0;
8      soma = soma + a;
9      soma = soma + b;
10     soma = soma + c;
11
12 }
```

O programa acima pode ser reescrito da seguinte maneira:

```
1  #include <stdio.h>
2
3  int a = 4, b = 6, c = 8;
4  int soma;
5
6  int main(){
7      soma = 0;
8      soma = soma + a;
9      soma = soma + b;
10     soma = soma + c;
11
12 }
```

```
1  #include <stdio.h>
2
3  int L[3] = {4,6,8};
4  int soma;
5
6  int main(){
7      soma = 0;
8      soma = soma + L[0];
9      soma = soma + L[1];
10     soma = soma + L[2];
11 }
```

No código à direita, em vez de declarar três variáveis inteiras separadamente, utilizamos um comando capaz de declarar uma lista com três elementos, denominados $L[0]$, $L[1]$ e $L[2]$. Além disso, como a única diferença entre os comandos das linhas 8, 9 e 10 é a posição do elemento utilizado, podemos substituí-los por um único comando que executa a mesma instrução para cada elemento, alterando apenas o elemento apontado na lista.

Utilizando o dedo

Chamaremos de dedo a variável que está sendo usada para apontar um elemento da lista.

```c
#include <stdio.h>

int L[3] = {4,6,8};
int soma;

int main(){
    soma = 0;
    soma = soma + L[0];
    soma = soma + L[1];
    soma = soma + L[2];
}
```

```c
#include <stdio.h>

int L[3] = {4,6,8};
int soma;

int main(){
    int i; //dedo
    soma = 0;

    for(i = 0; i <= 2; i++)
        soma = soma + L[i];
}
```

Podemos dizer que, nas linhas 10-11, estamos deslizando o dedo da posição 0 até a posição 2, avançando uma posição por vez e somando à variável soma o valor apontado pelo dedo i.

Maior3

```c
#include <stdio.h>

int L[3] = {4,6,8};
int maior;

int main(){
    maior = L[0];
    if( L[1] > maior) maior = L[1];
    if( L[2] > maior) maior = L[2];

}
```

```c
#include <stdio.h>

int L[3] = {4,6,8};
int maior;

int main(){
    int i; //dedo
    maior = L[0];

    for(i = 1; i <= 2; i++){
        if(L[i] > maior){
            maior = L[i];
        }
    }
}
```

Podemos dizer que, nas linhas 10-14, estamos deslizando o dedo da posição 1 até a posição 2, avançando uma posição por vez e se o valor L[i] supera o maior, atualizamos o valor do maior.

Maior e posição do maior

```c
#include <stdio.h>

int L[3] = {4,6,8};
int maior;
int pos_maior;

int main(){
    maior = L[0]; pos_maior = 0;
    if( L[1] > maior) {
        maior = L[1];
        pos_maior = 1;
    }
    if( L[2] > maior) {
        maior = L[2];
        pos_maior = 2;
    }

}
```

```c
#include <stdio.h>

int L[3] = {4,6,8};
int maior;

int main(){
    int i; //dedo
    maior = L[0];

    for(i = 1; i <= 2; i++)
        if(L[i] > maior){
            maior = L[i];
            pos_maior = i;
        }
}
```

Podemos dizer que, nas linhas 10-14, estamos deslizando o dedo da posição 1 até a posição 2, avançando uma posição por vez e se o valor L[i] supera o maior, atualizamos o valor do maior.

Busca4

```c
#include <stdio.h>

int L[4] = {4,6,8,10};
int x = 5;
int achei;

int main(){
    achei = 0;
    if(x == L[0]) achei = 1;
    else if(x == L[1]) achei = 1;
    else if(x == L[2]) achei = 1;

}
```

```c
#include <stdio.h>

int L[4] = {4,6,8, 10};
int x;
int achei;

int main(){
    int i; //dedo
    achei = 0;

    for(i = 0; i <= 2; i++)
        if(x == L[i]){
            achei = 1;
            break;
        }
}
```

Podemos dizer que, nas linhas 10-15, estamos deslizando o dedo da posição 0 até a posição 2, avançando uma posição por vez e se o valor x for igual ao valor de L[i], atualizamos o valor do achei para 1 e quebramos o laço.

Contando as cópias do maior

```c
#include <stdio.h>
#define N 10
int L[N] = { 5, 12, 3, 12, 8, 5, 10, 3, 12, 10 };
int maior;
int cont;

int main(){
    int i;
    // 1 Etapa: encontrar o maior
    maior = L[0];
    for(int i = 1; i < N; i++)
        if(L[i] > maior) maior = L[i];

    // 2 Etapa: contar o numero de ocorrencia
    cont  = 0;
    for(int i = 0; i < N; i++)
        if(L[i] == maior) cont++;
}
```

Nas linhas 11-12, deslizamos o dedo da posição 1 até a posição N-1, avançando uma posição por vez e se L[i] supera maior, atualizamos o valor de maior.

Nas linhas 16-17, deslizamos o dedo da posição 0 até a posição N-1, avançando uma posição por vez e se L[i] for igual ao maior, incrementamos o valor de cont.

```c
#include <stdio.h>
#define N 10
int L[N] = { 5, 12, 3, 12, 8, 5, 10, 3, 12, 10 };
int maior;
int cont;

int main(){
    int i;
    // 1 Etapa: encontrar o maior
    maior = L[0];
    cont = 1;
    for(int i = 1; i < N; i++)
        if(L[i] > maior) {
```

```
14              maior = L[i];
15              cont = 1;
16          }else if(L[i] == maior){
17              cont++;
18          }
19  }
```

Nas linhas 12-18, deslizamos o dedo da posição 1 até a posição N-1, avançando uma posição por vez e se L[i] superar o maior, atualizamos o maior e setamos cont igual a 1 senão se L[i] for igual a maior, incrementamos o valor de cont.