

Atividade 5 - Revisão e Listas

1 de outubro de 2025

Problema A

```
1 #include <stdio.h>
2
3 int buscaMenorElemento(int *vetor, int n){
4     if(n <= 0) return -1;
5
6     int menor = vetor[0];
7
8     for(int i = 0; i < n; ++i){
9         for(int j = i; j < n; ++j){
10             if(vetor[i] < menor){
11                 menor = vetor[i];
12             }
13         }
14     }
15
16     return menor;
17 }
18
19 int main() {
20     int n;
21     scanf("%d", &n);
22
23     int v[n];
24     for (int i = 0; i < n; i++) {
25         scanf("%d", &v[i]);
26     }
27
28     printf("Menor elemento: %d\n", buscaMenorElemento(v, n));
29
30     return 0;
31 }
```

Problema B

```
1 #include <stdio.h>
2
3 int contaPares(int *vetor, int n){
4     int contador = 0;
5
6     for(int i = 0; i < n; ++i){
7         for(int j = 0; j < n; ++j){
8             if(i == j){
9                 if(vetor[i] % 2 == 0){
10                     contador += 1;
11                 }
12             }
13         }
14     }
15
16     return contador;
17 }
18
19 // Complexidade = O(n)
20 int main(){
21     int n;
22     scanf("%d", &n);
23
24     int v[n];
25
26     for(int i = 0; i < n; ++i){
27         scanf("%d", &v[i]);
28     }
29
30     printf("%d\n", contaPares(v, n));
31
32     return 0;
33 }
```

Problema C

```
1 #include <stdio.h>
2
3 int contaTrios(int *vetor, int n, int soma){
4     int contador = 0;
5
6     for(int i = 0; i < n; ++i){
7         for(int j = i+1; j < n; ++j){
8             for(int k = j+1; k < n; ++k){
9                 if(vetor[i] + vetor[j] + vetor[k] == soma){
10                     contador += 1;
11                 }
12             }
13         }
14     }
15     return contador;
16 }
17
18 // Complexidade = O(n^3)
19 int main(){
20     int n, soma;
21     scanf("%d", &n);
22
23     int v[n];
24     for(int i = 0; i < n; ++i){
25         scanf("%d", &v[i]);
26     }
27     scanf("%d", &soma);
28     printf("%d\n", contaTrios(v, n, soma));
29     return 0;
30 }
```

Problema D

```
1 void InsereInicio(TipoItem x, TipoLista *Lista)
2 {
3     if(Lista->Ultio == MAXTAM) return;
4
5     for(TipoApontador p = Lista->Ultimo; p >= Lista->Primeiro; --p){
6         Lista->Item[p] = Lista->Item[p-1];
7     }
8     Lista->Item[Lista->Primeiro - 1] = x;
9
10    Lista->Ultimo += 1;
11 }
12
13 void RetiraPrimeiro(TipoLista *Lista, TipoItem *Item){
14     Retira(1, Lista, Item);
15 }
```