

Exame, 21 de Junho de 2022

1. Implemente **de forma eficiente** uma função `int pesquisa (int a[], int N, int x)` que, dado um **array ordenado** de tamanho $N > 0$, devolve um índice onde se encontra o valor x . Caso x não exista no array a função deverá devolver -1
2. Implemente uma função `void roda (LInt *l)` que move o último elemento da lista para a cabeça da mesma (sem alocar nova memória). Considere a definição usual do tipo `LInt`.

```
typedef struct LInt_nodo {  
    int valor;  
    struct LInt_nodo *prox;  
} *LInt;
```

3. Implemente uma função `int apaga (ABin a, int n)` que apaga n nodos de uma árvore binária. O critério para escolha de quais os nodos a apagar é livre. Se a árvore tiver menos do que n nodos então a deve apagar todos. A função deve devolver o número de nós efetivamente apagados. Considere a definição usual do tipo `ABin`.

```
typedef struct ABin_nodo {  
    int valor;  
    struct ABin_nodo *esq, *dir;  
} *ABin;
```

4. Implemente uma função `void checksum (char s[])` que, dada uma string s com um identificador só com dígitos, acrescenta-lhe um dígito de controle no final calculado de acordo com o método de Luhn. Neste método, o dígito de controle a incluir deve fazer com que a soma de todos os dígitos (incluindo o próprio dígito de controle) seja um múltiplo de 10. No entanto, no caso dos dígitos **em posições pares** (a começar do final) o que deve ser somado são os dígitos do número correspondente ao seu dobro. Por exemplo, dado o identificador "9871", a soma em questão corresponde a $9+1+6+7+2 = 25$ (note como o dígito 1 e 8 foram substituídos, respectivamente, por 2 e 1+6). Como a soma é 25, o dígito de controle a acrescentar deve ser 5, pelo que a string no final deverá ser "98715".
5. Implemente uma função `int escolhe (int N, int valor[], int peso[], int C, int quant[])` cujo objetivo é determinar a quantidade de produtos que um vendedor ambulante deve transportar. O vendedor tem à sua disposição uma quantidade ilimitada de N produtos diferentes, cujos valores e pesos estão guardados nos arrays `valor` e `peso`, respectivamente, mas só tem capacidade para

transportar C kg. A função deve tentar maximizar o valor total dos produtos a transportar, valor este que deve ser devolvido, e colocar no array quant a respectiva quantidade de cada produto. Por exemplo, se tivermos 3 produtos com valores $[20, 150, 30]$ e pesos $[2, 10, 3]$ e capacidade para 14 kg, então uma escolha ideal de quantidades seria $[2, 1, 0]$, correspondente ao valor total de 190. Mesmo que não consiga implementar uma estratégia de escolha óptima, implemente outra que ache razoável. O critério mais importante é o peso total não ultrapassar a capacidade de transporte C .