

Exame, 13 de Junho de 2023

1. Implemente a função `int isFib(int x)` que testa se um número `x` pertence à sequência de Fibonacci (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...).
2. Implemente a função `int moda(Aluno turma[], int N)` que, dado um array com a informação de `N` alunos, calcula a nota final mais frequente. Se houver mais do que uma nota final com a frequência máxima, devolva uma delas. Se a nota final mais frequente for “Reprovado” então deve devolver 0. O teste vale 80% da nota final e os mini-testes 20%.

```
typedef struct {  
    float teste, minis;  
} Aluno;
```

3. Apresente uma definição **iterativa** da função `int take(int n, LInt *l)` que, dado um inteiro `n` e uma lista ligada de inteiros `l`, apaga de `l` todos os nodos para além do `n`-ésimo (libertando o respectivo espaço). Se a lista tiver `n` ou menos nodos, a função não altera a lista. A função deve devolver o número de nodos apagados.

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

4. Apresente uma definição da função `int verifica(char frase[], int k)` que testa se todas as palavras que ocorrem numa frase têm pelo menos `k` caracteres.
5. Implemente a função `ABin reconstroi(char s[])` que dada uma string com uma string com uma travessia pré-order de uma árvore de dígitos, onde os apontadores nulos aparecem marcados com um '*', reconstrói a árvore original. Por exemplo, se a string for "34**52**5*6**" deverá ser devolvida a seguinte árvore.

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

