Programação Imperativa
Exame, 6 de Junho de 2024

Número: _____ Nome: _____

1. Implemente uma função `int pesquisa(int x, int a[], int N)` que, dado um array a ordenado de tamanho `N`, calcula em que **índice** desse array se encontra o elemento `x`. Se `x` não estiver no array deve devolver `-1`. Será valorizada a eficiência da função.

Programação Imperativa
Exame, 6 de Junho de 2024

Número: _____ Nome: _____

2. Implemente de forma iterativa uma função **void** duplicaTodos(**LInt** l) que duplica todos os elementos de uma lista. Se a lista tiver os valores [1,2,3] deverá ficar com os valores [1,1,2,2,3,3].

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

Programação Imperativa
Exame, 6 de Junho de 2024

Número: _____ Nome: _____

3. Implemente uma função `int expande(char s[])` que, dada uma string num formato compacto onde cada caracter é seguido de um dígito, expande essa string repetindo cada caracter o número de vezes indicado pelo dígito que o segue. A função deve devolver o tamanho da string expandida. Por exemplo, dada a string `"x3y1z4"` a função deve expandi-la para `"xxxyzzzz"`. Assuma que a string `s` tem espaço suficiente para armazenar a string expandida.

Programação Imperativa
Exame, 6 de Junho de 2024

Número: _____ Nome: _____

4. Adapte a função `fizzbuzz` definida abaixo por forma a devolver quantos números são impressos entre a primeira linha do tipo FizzBuzz e a última linha do tipo Buzz que se lhe segue. Se não aparecer nenhuma linha do tipo FizzBuzz ou nenhuma linha Buzz depois de uma linha do tipo FizzBuzz, então deve devolver `-1`.

```
int fizz(int n) {...}
int buzz(int n) {...}
int fizzbuzz(int n) {
     for (int i = 0; i < n; i++) {
          if (fizz(i) && buzz(i))
               printf("FizzBuzz\n");
          else if (fizz(i))
               printf("Fizz\n");
          else if (buzz(i))
               printf("Buzz\n");
          else
               printf("%d\n", i);
     }
     return 0;
}
```

Programação Imperativa
Exame, 6 de Junho de 2024

Número: _____ Nome: _____

5. Defina uma função **int** `todosIguais(`**ABin** `a,` **int** `n)` que testa se todos os valores no nível `n` de uma árvore são iguais (se não houver nenhum nodo a esse nível a função deve retornar verdadeiro). Serão valorizadas soluções que percorrem poucas vezes a árvore.

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

Programação Imperativa
Exame, 6 de Junho de 2024

Número: _____ Nome: _____

6. Uma expressão aritmética pode ser escrita em notação *reverse polish* passando os operadores para depois dos operandos. Nesse caso também já não são necessários parênteses para controlar a prioridade das operações. Por exemplo a expressão `"(3+4)*(2+1)"` seria escrita nesta notação como `"34+21+*"`. Implemente uma função `int calcula(char s[])`, que dada uma string com uma expressão na notação *reverse polish* (apenas com dígitos e os operadores + e *) calcula o seu valor. Por exemplo, `calcula("34+21+*") == 21`.