

1 Primeira questão (8 pontos)

Considere um array de caracteres de tamanho n qualquer.

1. Quantas permutações existem dos elementos do array em função de n ?
2. Complete o procedimento **recursivo** *permute* abaixo (3 linhas estão faltando), para que ele imprima todas as permutações dos elementos de um array.
3. O que o procedimento imprime com a chamada **permute(arr, 1, 3)**?
4. Desenhe uma interface para o Lazarus que permita entrar os elementos do array e exibir as permutações. Forneça os nomes de todos os componentes utilizados.

```
Type { programa para os items de 1 a 4 }
  ArrayType = Array[1..64] of char;
Var
  size: integer = 4;
  arr: ArrayType = '123456789';
procedure swap (var a: ArrayType; i, j: integer );
  var temp: char;
  begin
    temp := a[j];
    a[j] := a[i];
    a[i] := temp;
  end;
procedure permute (a: ArrayType; first, size: integer );
  var i: integer;
  begin
    if ( first = size ) then begin
      for i := 1 to size do
        write (a[i]);
        writeln
      end
    else begin
      for i := first to size do begin
        ...
        ...
        ...
      end
    end
  end;
end;
```

```
begin
```

```

    permute ( arr , 1 , size );
end .

```

a) $n!$

b)

```

    for i := first to size do begin
        swap ( a , first , i );
        permute ( a , first+1 , size );
        swap ( a , first , i );
    end

```

c)

```

123
132
213
231
321
312

```

d) Bastam 1 TEdit, 1 TButton e 1 ListBox (se não quiser imprimir na console).

2 Segunda questão (2 pontos)

O que o programa abaixo imprime, sabendo que a função **trim** remove todos os espaços em branco no início e no final de uma string? Use a letra 'b' para representar espaços em branco, se necessário.

```

Uses sysutils;
procedure seqR(s: String; c: String; i, added: integer);
begin
    if i > length(s) then begin
        if (length(trim(c)) > added) then
            writeln(c)
        end
    else begin
        seqR(s, c + s[i], i + 1, added + 1);
        seqR(s, c + ' ', i + 1, added)
    end
end;
begin
    seqR('1234', '', 1, 0);
end.

```

Saída do programa:

```

12 4
1 34
1 3
1 4
2 4

```