CORRIGE (partie théorique)

1. a) voir recueil « algorithmes obligatoires »
b) voir cours p. 80, 81 .
c) liste initiale : (' t ' , ' r ' , ' i ' , ' r ' , ' a ' , ' p ' , ' i ' , ' d ' , ' **e** ') (candidat = e)

(' d ' , ' r ' , ' i ' , ' r ' , ' a ' , ' p ' , ' i ' , ' t ' , ' **e** ')

(' d ' , ' a ' , ' i ' , ' r ' , ' r ' , ' p ' , ' i ' , ' t ' , ' **e** ')

liste finale: (' d ' , ' a ' , ' **e** ' , ' r ' , ' a ' , ' p ' , ' i ' , ' t ' , ' i ')

r

2. voir recueil « algorithmes obligatoires »

3. a) × examen1B_it ('2^4=4^2') : length(s) = 7

i	if s[i] <> s[length(s) – i + 1]	test
		true
1	false	true
2	false	true
3	false	true

D'où examen1B_it ('2^4=4^2') = true .

× examen1B_it ('2^6=4^3') : length(s) = 7

i	if s[i] <> s[length(s) – i + 1]	test
		true
1	true	false
2	false	false
3	true	false

D'où examen1B_it ('2^6=4^3') = false .

b) La fonction vérifie si une chaîne de caractères est symétrique ou non : si c'est le cas, l'ordre des lettres reste le même qu'on lise de gauche à droite ou de droite à gauche.

c)
```
function examen1B_rec(s:string) :boolean;
begin
    if length(s) <= 1 then result := true
    else if s[1]<>s[length(s)] then result := false
    else  result:=examen1B_rec(copy(s ,2 , length(s)-2))
end;
```