

### Question 1

→ le programme suivant tourne sur le Dauphin:

```
call  _WaitKey
Add   #3,A
Halt
```

On pèse sur la touche CTRL – 5 , quel est le contenu de la zone mémoire A lors de l'instruction Halt ?

- 1) H'23          2) H'45          3) 18          4) H'18
- 

### Question 2

→ Dans le programme qui génère la suite des nombres de Fibonacci (réaliser dans les exos libres), peut-on remplacer les instructions :

```
Push  Y
Pop   X
```

par :

```
Move  Y,A
Move  A,X
```

---

### Question 3

→ Parmi les affirmations suivantes, lesquelles sont correctes (plusieurs réponses peuvent correctes)?

- 1) Un microcontrôleur est un circuit intégré
  - 2) On ne peut programmer qu'une fois un microcontrôleur
  - 3) Un microcontrôleur est un microprocesseur plus moderne
  - 4) Un microcontrôleur est un système informatique
  - 5) Un microcontrôleur contient un microprocesseur
  - 6) Un microcontrôleur a toujours au moins 256 kB de RAM
- 

### Question 4

→ Parmi les affirmations suivantes, lesquelles sont correctes ?

- 1) Certains microcontrôleurs n'ont que 6 pattes
  - 2) Presque tous les fabricants de microcontrôleurs sont américains
  - 3) Les premiers microcontrôleurs sont apparus vers l'an 2000
  - 4) Certains les modèles de microcontrôleurs sont disponibles en boîtier Dual In Line (DIL, avec deux rangées de pattes coudées, qui peuvent passer dans les trous d'un circuit imprimé ou ceux qu'un BreadBoard)
  - 5) Plusieurs fabricants proposent des microcontrôleurs à base de processeurs ARM
  - 6) Tous les microcontrôleurs modernes peuvent être effacés par des ultra-violets
- 

### Question 5

→ Donnez une valeur approximative du nombre d'instructions élémentaires qu'un microcontrôleur est capable d'exécuter chaque seconde.

Vous pouvez donner une réponse qui correspond au microcontrôleur de votre choix.

## Question 6

→ Quelles lignes de code sont correctes du point de vue de leur syntaxe ?

- 1) → `delay (1000)`
  - 2) → `digitalWrite (3, 0);`
  - 3) → `if (digitalWrite (Pous) = 0)`
  - 4) → `while (PousOn) {delai (50);} // attend que l'on relâche`
  - 5) → `#define LedToggle digitalWrite (Led,!digitalRead(Led))`
  - 6) → `#define LedOff: digitalWrite (Led,HIGH)`
- 

## Question 7

→ On a 2 poussoirs et une LED. On veut allumer la LED avec le poussoir 1 et l'éteindre avec le poussoir 2. On hésite entre ces deux programmes :

```
// programme 1
void loop () {
    if (Pous1On) {LedOn;}
    if (Pous2On) {LedOff;}
}

// programme 2
void loop () {
    while (Pous1On) {LedOn;}
    while (Pous2On) {LedOff;}
}
```

Quelles sont les affirmations correctes ?

- 1) → Avec le programme 1 : quand on presse sur les deux poussoirs en même temps, je ne peux pas savoir ce qui se passe
  - 2) → Le programme 1 est correct
  - 3) → Avec le programme 1 : quand on presse sur les deux poussoirs en même temps, la LED clignote très rapidement
  - 4) → Le programme 2 est correct
  - 5) → Avec le programme 2 : quand on presse sur les deux poussoirs en même temps, la LED va être éteinte
  - 6) → Avec le programme 1 : quand on presse sur les deux poussoirs en même temps, la LED va être allumée
- 

## Question 8

→ On définit et initialise un compteur 8 bits (signed char cnt=7;). Dans le programme, on veut que le compteur décompte chaque seconde dans une boucle while (1) et se bloque à zéro.

Remarque : Le type *signed char* représente un nombre de 8 bits. Mais peut représenter des nombres positifs et négatifs (entre -128 et +127).

### Solution 1

```
while (1) {
    cnt--;
    if (cnt <0) {
        cnt++;
    }
    delay(1000);
}
```

**Solution 2**

```
while (1) {  
    if (cnt !=0) {  
        cnt--;  
    }  
    delay(1000);  
}
```

**Solution 3**

```
while (1) {  
    while (cnt !=0) {  
        cnt--;  
        delay(1000);  
    }  
}
```

Quelles sont les affirmations correctes ?

- 1) → La solution 2 pourrait être écrite de manière plus compacte :  

```
while (1) {  
    if (cnt !=0) { cnt--; }  
    delay(1000);  
}
```
- 2) → La solution 1 est correcte
- 3) → La solution 3 pourrait être écrite de manière plus compacte  

```
while (1) {  
    while (cnt !=0) { cnt--; }  
    delay(1000);  
}
```
- 4) → Les solutions 2 et 3 sont correctes