


Les fichiers .X68 et un compte rendu de TP contenant les organigrammes des programmes seront récupérés à la fin de la prochaine séance.

Exercice 1 :

Ecrire un programme qui utilise une boucle pour affiche "Bonjour 1", "Bonjour 2", ..., "Bonjour 10" à l'écran.

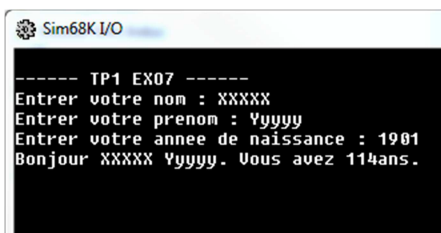


```

Sim68K I/O
----- TP1 EX06 -----
Bonjour 1
Bonjour 2
Bonjour 3
Bonjour 4
Bonjour 5
Bonjour 6
Bonjour 7
Bonjour 8
Bonjour 9
Bonjour 10
  
```

Exercice 2 :

Ecrire un programme qui demande à l'utilisateur de rentrer son nom, son prénom et son année de naissance et affiche après le nom suivi de prénom suivant de l'âge de l'utilisateur.



```

Sim68K I/O
----- TP1 EX07 -----
Entrez votre nom : XXXXX
Entrez votre prenom : Yyyyy
Entrez votre annee de naissance : 1901
Bonjour XXXXX Yyyyy. Vous avez 114ans.
  
```

Exercice 3 :

Ecrire un programme qui demande à l'utilisateur de saisir un nombre, ensuite le programme affiche si ce nombre est un nombre pair ou un nombre impair.

L'instruction « **DIVS Sr,Dn** » permet la division (des nombres signés) de **Dn** (32 bits) par **Sr** (16 bits) et met sur les 16 bits du poids plus fort de **Dn** le reste de la division et sur les 16 bits du poids le plus faible le résultat de la division.

L'instruction **SWAP** permet de permuter les bits de 16 à 31 et les bits 1 à 15.

Exemple :

```

MOVE #11,D1          D1 = 0 0 0 0 0 0 0 B
DIVS #4,D1            D1 = 0 0 0 3 0 0 0 2
                      Reste  Résultat de la division
  
```

Exercice 4 :

Adressages indirects pré/post décrémentés/incrémentés

Recopier une zone mémoire se trouvant entre les adresses \$2000 et \$2020.

1. Vers une destination \$2040
 - a. Recopier octet par octet.
 - b. Recopier mot par mot.
 - c. Recopier mot long par mot long.
2. Vers une destination \$2041, que constate-t-on ?