

I. Trier une liste de mots de 32 bits non signés à l'aide d'une procédure **Trisel(N, AdrListe)**.

a- Ecrire un programme qui génère une liste de N nombres de 32 bits non signés ($9 < N < 100$). Les nombres sont stockés dans une liste (L).

b- Construire la procédure et écrire un programme de test qui trie une petite liste **L** de **N** mots de 32 bits non signés prédéfinie dans la section des données.

Le nombre d'éléments sera transmis par valeur et le second paramètre sera l'adresse de la liste. Le passage des paramètres sera effectué en utilisant la pile.

II. Utiliser la conversion hexadécimal vers ASCII du cours pour créer une chaîne de caractères A Zéro Terminal représentant une **petite** liste triée d'au plus 10 nombres.

La procédure **ConvHexA(V, AdrChaine)** place à l'adresse **AdrChaine** la séquence de 8 caractères représentant **V** en hexadécimal. Les paramètres V et AdrChaine seront transmis dans R0 et R1 (convention C pour IAR kit).

Annexes

Utilisation de la pile par les procédures

Une **procédure appelante** peut utiliser la pile pour le passage des paramètres.

Elle empile la valeur ou l'adresse d'un paramètre suivant le mode de transmission.

Puis elle appelle le sous-programme avec l'instruction **BL SousProg**.

La **procédure appelée** a ensuite quatre tâches à effectuer :

- Sauvegarder **LR**, dans la pile, si la procédure appelle un sous-programme.
- Empiler un registre **Rp** et l'initialiser avec la valeur de **SP**, pour préparer l'accès aux paramètres à l'aide d'un pointeur fixe. Les paramètres et les variables locales sont repérés par un déplacement relatif à Rp.
- Sauvegarder les registres qu'elle ne doit pas modifier, si une convention prévoit qu'elle doit assumer cette tâche.
- Retrancher la taille des variables locales à SP (SP doit avoir une valeur multiple de 4) pour allouer une zone de pile aux variables locales.

Elle peut ensuite exécuter le traitement pour lequel elle est conçue. La partie supérieure de la pile est utilisée "normalement".

En fin d'exécution, elle effectue les opérations inverses : libérer les variables locales ; dépiler les registres sauvés, Rp et LR.

Elle peut aussi libérer la zone des paramètres en ajoutant *n* à SP, où *n* est la taille de cette zone en octets. Le retour est assuré par l'instruction **BX LR** ou **MOV PC, LR**.

Une fonction retourne son résultat dans un ou plusieurs registres.

Des procédures dédiées ou locales à une autre, peuvent avoir un prologue et un épilogue simplifiés et optimisés en fonction des traitements effectués.

On peut faire la **mise au point** à l'aide de « Run to cursor » ou avec un double clic dans la marge grise pour placer des points d'arrêt puis « Go ».