

## Oefeningenreeks 4: Herhalingsoefening + Van C naar Assembleertaal

Bij het omzetten van C naar assembleertaal is het belangrijk om te weten of je een programma moet schrijven dan wel een subroutine. Bij een subroutine moet voor de aanroep de volledige CPU-context (i.e. alle werkregisters dus) naar de stack worden gekopieerd en op het einde van de subroutine terug worden gehaald. Het kan ook eenvoudiger door bv. enkel de registers die binnen de subroutine gebruikt worden te bewaren en later terug te zetten. Het kan nooit de bedoeling zijn dat een subroutine een werkregister van inhoud wijzigt.

Zowel voor het schrijven van een programma als voor het schrijven van een subroutine moet je ook weten dat lokale variabelen zich op de stapel bevinden.

1. Schrijf een ASM-programma dat alle cijfers tussen [0-15] in hexadecimale vorm op een 7-segmentdisplay toont. Iedere 10 seconden wordt het volgende cijfer getoond. Gebruik gerust een omzettabel waar index 0 zich op adres 40H bevindt. Gebruik ook interrupts. Bij het indrukken van de knop die verbonden is met INT0 wordt er onmiddellijk gestopt.
2. Zet onderstaande code letterlijk om naar assembleertaal:

```
int main(){
    int tab[]={1,2,3,7,15,0x23};
    int i=0;
    while (i<6 && tab[i]!=7) i++;
    P1.6=(i==6?1:0);
    while(1);
}
```

3. Schrijf een kleine functie `vermenigvuldig(a,b)` die twee parameters `a` en `b` neemt (die worden bij het aanroepen via de stapel doorgegeven). De functie geeft de LSB van het product terug. (De teruggeefwaarde bevindt zich steeds in de accumulator).
4. Schrijf een functie `geef_kleinste(a,b,c)` die het kleinste element bepaalt en dit teruggeeft. De paramaters worden opnieuw doorgegeven via de stack. De returnwaarde bevindt zich ook zoals steeds in de accumulator.
5. Schrijf nu een functie `geef_laatste_element (array,n)` die het element op de laatste index van een array teruggeeft.

6. De bedoeling is nu om het zoeken van een getal in een subroutine te gaan onderbrengen. We krijgen nu de volgende code:

```
int zoek(const int *tab, int n, int getal){
    int i=0;
    while(i<n && tab[i]!=getal) i++;
    if (i==n) return 0;
    else return 1;
}
```

```
int main(){
    int tab[]={1,2,3,7,15,0x23};
    if (zoek(tab,6,7)==1)
        P1.6=0; //aan dus
    else
        P1.6=1; //uit dus
    while(1);
}
```

7. Zet onderstaande code letterlijk om naar assembleertaal:

```
void wissel (int *a, int *b){
    int hulp=*a;
    *a=*b;
    *b=hulp;
    return;
}

int main(){
    int x=17;
    int y=15;
    wissel(&x,&y);
    while(1);
}
```