ΠΑΝ ΔΥΤ. ΜΑΚΕΔΟΝΙΑΣ **ΤΜΗΜΑ ΜΗΧ. ΠΛΗΡΟΦΟΡΙΚΗΣ κ ΤΛΠ.**
ΧΕΙΜ. ΕΞΑΜ. 2014-2015 ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ
Διδάσκων: Κοντογιάννης Σωτήριος

ΕΡΓΑΣΤΗΡΙΟ 4

ΑΣΚΗΣΗ 1: Να δημιουργήσετε ένα πρόγραμμα σε γλώσσα assembly που θα δηλώνει ένα 16bit μετρητή. Στη συνέχεια να δημιουργηθεί υπορουτίνα που θα μειώνει αυτόν τον μετρητή κατά 1 και μία υπορουτίνα που θα αυξάνει τον μετρητή κατά ένα. Να κληθούν οι ρουτίνες και το αποτέλεσμα να εμφανιστεί με καθυστέρηση ενός δευτερολέπτου στη σειριακή θύρα.

ΑΣΚΗΣΗ 2: Να δημιουργήσετε ένα πρόγραμμα σε assembly το οποίο θα δέχεται ως είσοδο ένα χαρακτήρα, π.χ. char ch='A'; και θα αυξάνει την τιμή του κατά ένα χαρακτήρα στον πίνακα ASCII (ch='A'+1='B').

ΑΣΚΗΣΗ 3: Να τροποποιήσετε το παραπάνω πρόγραμμα ώστε να χρησιμοποιεί υπορρουτίνα με χρήση του STACK για τον απαριθμητή χαρακτήρων.

ΑΣΚΗΣΗ 4: Να δημιουργήσετε ένα πρόγραμμα σε assembly το οποίο θα περιέχει υπορουτίνα που θα υπολογίζει το αποτέλεσμα της γραμμικής συνάρτησης ακεραίων 1 byte θετικών αριθμών y=ax+b, όπου a,b,x δίνονται από το πληκτρολόγιο.

ΑΣΚΗΣΗ 5: Να τροποποιήσετε το παραπάνω πρόγραμμα ώστε να χρησιμοποιεί υπορρουτίνα με χρήση του STACK για τον υπολογισμό της γραμμικής συνάρτησης.

<u>Υπόδειγμα 1.</u>

ADD	Rd, Rr	Add without Carry	Rd ← Rd + Rr	Z,C,N,V,S,H	1
ADC	Rd, Rr	Add with Carry	Rd ← Rd + Rr + C	Z,C,N,V,S,H	1
ADIW	Rd, K	Add Immediate to Word	Rd+1:Rd ← Rd+1:Rd + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract without Carry	Rd ← Rd - Rr	Z,C,N,V,S,H	1
SUBI	Rd, K	Subtract Immediate	Rd ← Rd - K	Z,C,N,V,S,H	1
SBC	Rd, Rr	Subtract with Carry	Rd ← Rd - Rr - C	Z,C,N,V,S,H	1
SBCI	Rd, K	Subtract Immediate with Carry	Rd ← Rd - K - C	Z,C,N,V,S,H	1
SBIW	Rd, K	Subtract Immediate from Word	Rd+1:Rd ← Rd+1:Rd - K	Z,C,N,V,S	2

```
#include "asmtest.h"

void setup() {
Serial.begin(9600);
count=0x01FF;
//asm("call myinit");
 //smycall();
}

void loop()
{

  Serial.println(count,HEX);
  delay(1000);
  cup();

}
```

```
#include <avr/io.h>
#devine __SFR_OFFSET    0
#include "asmtest.h"
.section .data
.extern count
.section .text
.global cup
cup:
clr r24
clr r25
clr r23
clr r22
lds r24,count
lds r25,count+1
ldi r23,0x01
ldi r22,0x00
call myadd
ret
myadd:
//adiw r24,0x0001
//sts count,r24
//sts count+1,r25
adc r25,r22 ;Add low byte
adc r24,r23 ;ADD Carry if set in status register high byte
sts count,r24
sts count+1,r25
ret

.end
```

```
/*
 * Global register variables.
 */
#ifdef __ASSEMBLER__

/* Assembler-only stuff */

#else  /* !ASSEMBLER */

/* C-only stuff */

#include <stdint.h>
char s1='A';
extern "C" void stack_init(void);
extern "C" void stack_add(void);
#endif /* ASSEMBLER */
```

Υπόδειγμα 2/3.

```
/*
 * Global register variables.
 */
#ifdef __ASSEMBLER__

/* Assembler-only stuff */

#else  /* !ASSEMBLER */

/* C-only stuff */

#include <stdint.h>
char sl='A';
extern "C" void stack_init(void);
extern "C" void stack_add(void);
#endif /* ASSEMBLER */
```

```
#define __SFR_OFFSET 0
/*Automatically subtract 0x20 from I/O addresses
*/
#include <avr/io.h>
.section .data
message:
.extern sl;
.section .text
.global stack_add
stack_add:
lds r16,sl;
push r0;
push r16;
call m_stack_add
pop r0
pop r18
sts sl,r18
m_stack_add:
IN R31,SPH//SPH=0x3E
IN R30,SPL //SPL=0x3D
LDD r18,Z+3
LDD r19,Z+4
INC r18
STD Z+4,r18
OUT 0x3E,R31
OUT 0x3D,R30
ret
.end
```

```
#include "asmtest.h"
#include "WString.h"

void setup() {
//stack_init(); /*Use always the same stack---NEVER INIT!!!*/
Serial.begin(9600);

Serial.println(sl);
stack_add();
}

void loop()
{
  Serial.println(sl);
  delay(1000);
}
```

Υπόδειγμα 4/5.

```
#include "asmtest.h"

void setup() {
//stack_init(); /*Use always the same stack---NEVER INIT!!!*/
Serial.begin(9600);
while (!Serial) {}
  delay(2000);
  Serial.print("a=");
  Serial.setTimeout(5000);
  ma=Serial.parseInt();
  Serial.println(ma);
  Serial.print("b=");
  Serial.setTimeout(5000);
   mb=Serial.parseInt();
  Serial.println(mb);
}
void loop()
{
 mlinear();
 Serial.print("Result=");
  Serial.println(my);
  mx++;
  delay(3000);
}
```

```
/*
 * Global register variables.
 */
#ifdef __ASSEMBLER__

/* Assembler-only stuff */

#else  /* !ASSEMBLER */

/* C-only stuff */

#include <stdint.h>
extern uint8_t ma;
extern uint8_t mb;
extern uint8_t mx; //counter
extern uint8_t my;
extern "C" void mlinear(void);
#endif /* ASSEMBLER */
```

```
#define __SFR_OFFSET 0
/*Automatically subtract 0x20 from I/O addresses
*/
#include <avr/io.h>
.section .data
.global ma
ma: .byte 0x00;
.global mb
mb: .byte 0x00;
.global mx
mx: .byte 0x01;
.global my
my: .byte 0x00;
.section .text
.global mlinear
mlinear:
lds r25,mx;
lds r23,ma;
lds r24,mb;
push r0; //result
push r23;
push r24;
push r25;
call m_stack_lin
pop r0
pop r0;
pop r0;
pop r23;
sts my,r23
ret
```

```
m_stack_lin:
IN R31,SPH//SPH=0x3E
IN R30,SPL //SPL=0x3D
LDD r18,Z+3 //mx
LDD r19,Z+4 //b
LDD r20,Z+5 //a
LDD r22,Z+6 //my
MUL r20,r18
MOV r20,r0
add r20,r19
STD Z+6,r20
OUT SPH,R31
OUT SPL,R30
ret
.end
```