

Design Assignment 1B

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

The student understands that all required components should be submitted in complete for grading of this assignment.

NO	SUBMISSION ITEM	COMPLETED (Y/N)	MARKS (/MAX)
1	COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS		
2.	INITIAL CODE OF TASK 1/A		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E		
4.	SCHEMATICS		
5.	SCREENSHOTS OF EACH TASK OUTPUT		
5.	SCREENSHOT OF EACH DEMO		
6.	VIDEO LINKS OF EACH DEMO		
7.	GOOGLECODE LINK OF THE DA		

1.	INITIAL CODE		
----	--------------	--	--

```
.org 0
ldi R21, 0
ldi XL, 0x00      ;X to 0x0200
ldi XH, 0x02
ldi YL, 0x00      ;Y to 0x0400
ldi YH, 0x04
ldi ZL, 0x00      ;Z to 0x0600
ldi ZH, 0x06
```

FILL:

```
ldi R20, 0
add R20, XL
add R20, XH
st X, R20
inc XL
cpi XL, 255
brsh RESETX
```

CONTINUE:

```
inc R21
jmp DIVIDE
```

2.	Code to check if value is divisible by 5		
----	--	--	--

DIVIDE:

```
ldi R22, 3
ldi R23, 0
mov R24, R20
```

L1:

```
inc R23
sub R20, R22
BRCC L1
dec R23
add R20, R22
cpi R20, 0
brne NOTDIV
st Y, R24
inc YL
ldi R25, 0
```

add R16, R24

adc R17, R15

jmp DIVIDERET

3.	Sum values in R16:R17 also R18:R19		
----	------------------------------------	--	--

```
st Y, R24
inc YL
ldi R25, 0
add R16, R24
adc R17, R15
jmp DIVIDERET
```

NOTDIV:

```
st Z, R24
inc ZL
ldi R25, 0
add R18, R24
adc R19, R15
jmp DIVIDERET
```

4.	Complete code		
----	---------------	--	--

```
;
; DA1B.asm
;
; Created: 2/20/2019 9:53:23 AM
; Author : YKengne
;
```

```
.org 0
ldi R21, 0
ldi XL, 0x00      ;X to 0x0200
ldi XH, 0x02
ldi YL, 0x00      ;Y to 0x0400
ldi YH, 0x04
ldi ZL, 0x00      ;Z to 0x0600
ldi ZH, 0x06
```

FILL:

```
ldi R20, 0
add R20, XL
add R20, XH
st X, R20
inc XL
cpi XL, 255
brsh RESETX
```

CONTINUE:

```
inc R21
jmp DIVIDE
```

DIVIDERET:

```
cpi R21, 255
brne FILL
```

```

        ldi R21, 0
FILLMORE:
        ldi R20, 0
        add R20, XL
        add R20, XH
        st X, R20
        inc XL
        inc R21
        jmp DIVIDE2
DIVIDERET2:
        cpi R21, 42
        brne FILLMORE
        jmp END

```

```

RESETX:
        ldi XL, 0
        inc XH
        jmp CONTINUE

```

```

DIVIDE:
        ldi R22, 3
        ldi R23, 0
        mov R24, R20
L1:
        inc R23
        sub R20, R22
        BRCC L1
        dec R23
        add R20, R22
        cpi R20, 0
        brne NOTDIV
        st Y, R24
        inc YL
        ldi R25, 0
        add R16, R24
        adc R17, R15
        jmp DIVIDERET

```

```

NOTDIV:
        st Z, R24
        inc ZL
        ldi R25, 0
        add R18, R24
        adc R19, R15
        jmp DIVIDERET

```

```

DIVIDE2:
        ldi R22, 3
        ldi R23, 0
        mov R24, R20
L2:
        inc R23
        sub R20, R22
        BRCC L2
        dec R23
        add R20, R22
        cpi R20, 0
        brne NOTDIV2

```

```

st Y, R24
inc YL
ldi R25, 0
add R16, R24
adc R17, R25
jmp DIVIDERET2

```

NOTDIV2:

```

st Z, R24
inc ZL
ldi R25, 0
add R18, R24
add R19, R25
jmp DIVIDERET2

```

END:

```

jmp END

```

5.	Screenshots of each task		
----	--------------------------	--	--

Task1

Memory 4

Memory: data REGISTERS Address: 0x0200,data

Address	Hex	ASCII
0x0200	02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13
0x0212	14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 !"#\$%
0x0224	26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37	&'()*+,-./01234567
0x0236	38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49	89:;<=>?@ABCDEFGHI
0x0248	4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 59 5a 5b	JKLMNOPQRSTUVWXYZ[
0x025A	5c 5d 5e 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d	\]^_`abcdefghijklmnopqrstuvwxyz{ }~.
0x026C	6e 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f	€..f....^..Š.Œ.Ž..‘
0x027E	80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f 90 91	’“”.—™š.œ.žŸ ¡¢£
0x0290	92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3	¥!§““«¬.®~°±..´μ
0x02A2	a4 a5 a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5	¶·.º»...¿ÀÁÂÃÄÅÆÇ
0x02B4	b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5 c6 c7	

Call Stack Breakpoints Command Window Immediate Window Output Memory 4

Task2

Memory 4																																						
Memory:	data REGISTERS															Address:	0x0400,data																					
data 0x0200	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	10	11	12	13																			
data 0x0212	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f	20	21	22	23	24	25 !"#\$%																			
data 0x0224	26	27	28	29	2a	2b	2c	2d	2e	2f	30	31	32	33	34	35	36	37	&'()*+,-./01234567																			
data 0x0236	38	39	3a	3b	3c	3d	3e	3f	40	41	42	43	44	45	46	47	48	49	89:;<=>?@ABCDEFGHI																			
data 0x0248	4a	4b	4c	4d	4e	4f	50	51	52	53	54	55	56	57	58	59	5a	5b	JKLMNOPQRSTUVWXYZ[
data 0x025A	5c	5d	5e	5f	60	61	62	63	64	65	66	67	68	69	6a	6b	6c	6d	\]^_`abcdefghijklmnopqrstuvwxyz{ }~.																			
data 0x026C	6e	6f	70	71	72	73	74	75	76	77	78	79	7a	7b	7c	7d	7e	7f	nopqrstuvwxyz{ }~.																			
data 0x027E	80	81	82	83	84	85	86	87	88	89	8a	8b	8c	8d	8e	8f	90	91	€..f....^..Š.Ě.Ž..‘																			
data 0x0290	92	93	94	95	96	97	98	99	9a	9b	9c	9d	9e	9f	a0	a1	a2	a3	’“”...—™š.œ.žŸ ¡¢£																			
data 0x02A2	a4	a5	a6	a7	a8	a9	aa	ab	ac	ad	ae	af	b0	b1	b2	b3	b4	b5	¤¥¦§¨©ª«¬.®¯°±..´μ																			
data 0x02B4	b6	b7	b8	b9	ba	bb	bc	bd	be	bf	c0	c1	c2	c3	c4	c5	c6	c7	¶·¸¹º»...¿ÀÁÂÃÄÅÆÇ																			
Call Stack Breakpoints Command Window Immediate Window Output Memory 4																																						

Task3

Memory 4

Memory: data REGISTERS

Address: 0x0600,data

data 0x0200	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	10	11	12	13
data 0x0212	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f	20	21	22	23	24	25 !"#%
data 0x0224	26	27	28	29	2a	2b	2c	2d	2e	2f	30	31	32	33	34	35	36	37	&'()*+,-./01234567
data 0x0236	38	39	3a	3b	3c	3d	3e	3f	40	41	42	43	44	45	46	47	48	49	89:;<=>?@ABCDEFGHI
data 0x0248	4a	4b	4c	4d	4e	4f	50	51	52	53	54	55	56	57	58	59	5a	5b	J K L M N O P Q R S T U V W X Y Z [
data 0x025A	5c	5d	5e	5f	60	61	62	63	64	65	66	67	68	69	6a	6b	6c	6d	\] ^ _ ` a b c d e f g h i j k l m
data 0x026C	6e	6f	70	71	72	73	74	75	76	77	78	79	7a	7b	7c	7d	7e	7f	n o p q r s t u v w x y z { } ~ .
data 0x027E	80	81	82	83	84	85	86	87	88	89	8a	8b	8c	8d	8e	8f	90	91	€ . . f ^ . . Š . Ě . Ž . . ‘
data 0x0290	92	93	94	95	96	97	98	99	9a	9b	9c	9d	9e	9f	a0	a1	a2	a3	’ “ ” . . — ™ š . œ . ž Ÿ ¡ ¢ £
data 0x02A2	a4	a5	a6	a7	a8	a9	aa	ab	ac	ad	ae	af	b0	b1	b2	b3	b4	b5	¤ ¥ ¦ § ¨ © ª « ¬ . ® ¯ ° ± . . ´ μ
data 0x02B4	b6	b7	b8	b9	ba	bb	bc	bd	be	bf	c0	c1	c2	c3	c4	c5	c6	c7	¶ · ¸ ¹ º » . . . ¿ À Á Â Ã Ä Å Æ Ç

Call Stack Breakpoints Command Window Immediate Window Output Memory 4

C CODE

```
#include <avr/io.h>
```

```
int main()
{
    int *X = 0x0200;
    int *Y = 0x0400;
    int *Z = 0x0600;
    int a = 0;
    char sum = 0;
    char R16;
    char R17;
    char R18;
    char R19;

    for(a=0; a<99; a++)
    {
```

```
sum = 0;
sum = X;
*X = sum;
if(sum/3 == 0)
{
    *Y = sum;
    Y++;
    R16 = sum;
    R17 = sum;
}
else
{
    *Z = sum;
    Z++;
    R18 = sum;
    R19 = sum;
}
X++;
}
}
```

Name	Value
Program Counter	0x00000040
Stack Pointer	0x08FD
X Register	0x0000
Y Register	0x08FF
Z Register	0x0000
Status Register	1 1 1 1 1 1 1 1
Cycle Counter	12
Frequency	16.000 MHz
Stop Watch	0.75 μ s
Registers	
R00	0x00
R01	0x00
R02	0x00
R03	0x00
R04	0x00
R05	0x00
R06	0x00
R07	0x00
R08	0x00
R09	0x00
R10	0x00
R11	0x00
R12	0x00
R13	0x00
R14	0x00
R15	0x00
R16	0x00
R17	0x00
R18	0x00
R19	0x00
R20	0x00
R21	0x00
R22	0x00
R23	0x00
R24	0x00
R25	0x00
R26	0x00

2.	GITHUB LINK OF THE DA		
https://github.com/Vasty1995/submission_da			

Student Academic Misconduct Policy

<http://studentconduct.unlv.edu/misconduct/policy.html>

"This assignment submission is my own, original work".
Yannick Kengne Tatcha