CPE301 – SPRING 2019

Design Assignment 1B

Student Name: Alfonso Contreras

Student #: 1012383150

Student Email: contra2@unlv.nevada.edu

Primary Github address: https://github.com/alf8420/Submit.git

Directory: alf8420/Submit

Submit the following for all Labs:

1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

**N/A**

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**
2. ;
3. ; AssemblerApplication1.asm
4. ;
5. ; Created: 2/23/2019 1:13:04 AM
6. ; Author : ALFONSO CONTRERAS
7. ; EMAIL: CONTRA2@UNLV.NEVADA.EDU
8. ;
9. ; Replace with your application code
10. COUNTERS:
11. .equ STARTADDS= 0X0200 ; Starting Address
12. .equ counter1= 255 ; FIRST COUNTER
13. .equ counter2=10 ; SECOND COUNTER
14. .org 0
15. CLR R0 ; Clearing RO
17. LDI XL, LOW(STARTADDS) ; STARTADDS[11:0]
18. LDI XH, HIGH(STARTADDS) ; STARTADDS[255:0]
20. LDI YL, LOW(0X400) ; FOR DIVISIBLE #'S BY 3
21. LDI YL, HIGH(0X400)
22. LDI ZL, LOW(0X600) ; REST OF #'S
23. LDI ZH, HIGH(0X600)
24. LDI R23, COUNTER1 ; R23=255
25. LDI R24, COUNTER2 ; R24=10
26. ; CLEANING REGISTERS TO ZERO
27. CLR R0=0 ; R0=0
28. CLR R3=0 ; R3=0
29. CLR R4=0 ; R4=0
30. CLR R6=0 ; R6=0
31. CLR R7=0 ; R7=0
32. CLR R16=0 ; R16=0
33. CLR R17=0 ; R17=0
34. CLR R18=0 ; R18=0
35. CLR R19=0 ; R19=0
36. BEGIN:
37. MOV R3, XL ; R3 TAKES LOW VALUES
38. ADD R3, XH ; R3 TAKES ADDITION OF HIGH AND LOW
39. MOV R4, R3 ; COPY R3 INTO R4
40. ST X+, R4 ; INCREMENTING MY POINTER
41. LDI R6, 3 ;LOADING R6=3 TO SEE IF THE #'S DIVISIBLE BY 3
43. INC R7 ; INCREMENTING R7
44. DIVBYTHREE:
45. CPI R3, R6 ; COMPARE IMMEDIATELY R6 WITH R3
46. BRLO DIV\_NO\_BUENO ; IF IS NOT DIVISIBLE BY 3
47. SUB R3, R6 ; R3 MINUS R6
48. CPI R3, R0 ; IF EQUAL TO ZERO
49. BREQ DIV\_BUENO ; GO TO DIV\_BUENO
50. DIV\_BUENO:
51. ST Z+, R4 ; INCREMENT LOW AND HIGH VALUES OF Z
52. ADD R16, R4 ; ADD R4 TO R16 LOW VALUES
53. ADC R17, R0 ; R17= R0+R17
54. RJMP MADE ; GO TO MADE

57. MADE:
58. CPI R23, RO ; IF COUNTER EQUAL ZERO
59. BREQ ALMOST ; GO TO ALMOST
60. SUB R23, R7 ; SUBTRACT R7 FROM R23
61. BRNE BEGIN ; IF R23 IS NOT ZERO
62. ALMOST:
63. SUB R24, R7; ; R24 MINUS R7
64. BRNE BEGIN ; IF R24 IS NOT ZERO
65. END:
66. RJMP COUNTERS ; COUNTERS
68. .ORG 0x20
69. STARTADDS:
70. .DB 11, 12,13,14,15,16,17,18,19,20,25, 26,29,41,43,56,58,68,39,22,41, 42,73,84,85,96,87,88,89,90,35, 36,39,51,53,66,68,78,79,12,61, 112,113,114,115,116,117,118,119,200,205, 206,209,40,44,46,48,68,69,62,111, 122,173,184,185,196,187,188,189,190,135, 136,139,151,153,166,168,178,179,112,110, 222,223,224,225,216,217,218,219,220,125, 226,219,141,243,215,158,168,139
71. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

**C++ code to prove is working.**

#include <iostream>

#include <random>

#include <iomanip>

using namespace std;

const int MAX = 100;

int \*X[MAX]; //Pointer X to populate 1-99 numbers

int \*Y[MAX]; // Pointer Y to save numbers divisible by 3

int \*Z[MAX]; // Pointer Z to save the rest of the numbers

int nodiv3 = 0;

int restnum = 0;

int main()

{

random\_device rd; //variable used for random

int g[MAX]; //array used to populate my pointers

for (int i = 0; i < MAX; i++)

{

g[i] = 0;

}

//random generator function

mt19937 gen(rd());

// uniformly distributed in range (10, 255)

uniform\_int\_distribution<> dis(10, 255);

// Generate 100 random numbers and populate my array

for (int i = 0; i < MAX; i++)

{

int randomX = dis(gen);

g[i] = randomX;

X[i] = &g[i];

}

//Populate my Y array, which will serve for numbers divisible by 3

for (int i = 0; i < MAX; i++)

{

int randomX = dis(gen);

g[i] = randomX;

//if is divisible by 3

if (g[i] % 3 == 0)

{

Y[i] = &g[i];

nodiv3++;

}

else

{

Z[i] = &g[i];

restnum++;

}

}

//Print the whole numbers

for (int i = 0; i < MAX; i++)

{

cout <<right<<setw(6)<< "Random " << i << " X: " << \*X[i] << endl;

}

for (int j = 0; j < nodiv3; j++)

{

//Pointers equal to zero, don't care xxx

if (Y[j] != 0)

{

cout <<setw(14)<< "Random Y:" << j << " = " << \*Y[j] << endl;

}

}

//Print the numbers there not divisible by 3

for (int l = 0; l < restnum; l++)

{

if (Z[l] != 0)

{

cout << setw(20) << "Random Z:" << l << " = " << \*Z[l] << endl;

}

}

int total = 0;

//Calculate total

for (int k = 0; k < MAX; k++)

{

total = total+ \*X[k];

}

//Print my total

cout << setw(30) << "total= " << " = " << total << endl;

system("pause");

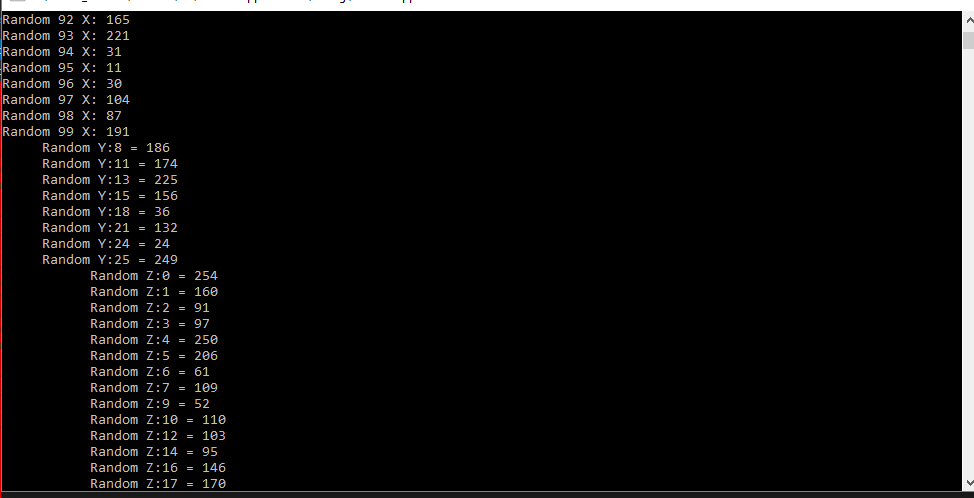
return 0;

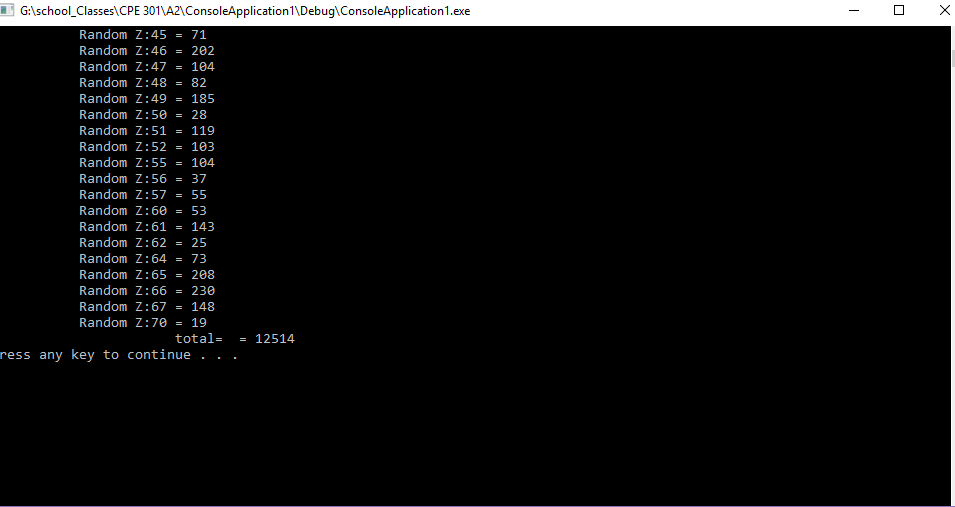
delete X;

delete Y;

delete Z;

}

****

****

1. **SCHEMATICS**

N/A

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**
2. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**
3. **VIDEO LINKS OF EACH DEMO**
4. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

“This assignment submission is my own, original work”.

NAME OF THE STUDENT