CS 218
Homework, Asst. #2

Purpose: Become familiar with the assembler, linker, and debugger.
Display values in memory for integers, reals, and characters.
Due: Thursday (6/09), Before 1:00 PM
Points: 20 (Part A - 10pts, Part B - 10pts)

Assignment:

Part A:

Write a simple assembly language program to compute the following formulas:

```
bAns1  =  bVar1 + bVar2
bAns2  =  bVar1 - bVar2
wAns1  =  wVar1 + wVar2
wAns2  =  wVar1 - wVar2
dAns1  =  dVar1 + dVar2
dAns2  =  dVar1 - dVar2
```

Declare the following variables in the data segment (after the ".data").

```
NULL            equ    0

bVar1           db     31
bVar2           db     13
bAns1           db     0
bAns2           db     0
wVar1           dw     3924
wVar2           dw     1943
wAns1           dw     0
wAns2           dw     0
dVar1           dd     174632105
dVar2           dd     124789360
dVar3           dd     -56237
dAns1           dd     0
dAns2           dd     0
flt1            dd     -14.25
flt2            dd     7.125
threePi         dd     9.42477
qVar1           dq     123169427153
myClass         db     "CS 218", NULL
edName          db     "Ed Jorgensen", NULL
myName          db     "your name goes here", NULL
```

Submission

1) Assemble the program, ***print and submit the assembler list file*** (i.e., ast2.lst).
2) Using the debugger, execute the program. Use the provided debugger display commands ("a2in.txt"). ***Print and submit a copy of the values after execution*** (i.e., a2out.txt).

Part B:
On the attached worksheet, write the binary, decimal, and hex representation of each integer variable. For the reals (*flt1, flt2*, and *threePi*), in addition, show the sign, exponent, and mantissa parts of the representation. For the character strings, show the ASCII codes (for each letter) in hex for only the first six (6) characters for each. See the worksheet for examples.

Submission:
Submit a hard copy of:

1) the assembler list, which is created after the assemble command (i.e., "ast2.lst")
2) debugger printout of the all the variables (i.e, "a2out.txt")
3) completed data representation worksheet.

Note, the assignment is due at the beginning of class.

Debugger Commands:
Execute the program in the debugger (in the same manner as assignment #1). You should review the DDD/GDB debugger information handout to understand the debugger commands examine memory variables.

You should use the provided "a2in.txt" to display the variables with the debugger.

- Each byte, word, double-word sized, and quadword variable is displayed twice (once in decimal and again in hex).
- The floating point values are display twice (once as a real value and again in hex).
- The strings are displayed twice, once showing both the decimal and ASCII values and then just the hex values for the first six characters

A brief summary of the command to examine memory is as follows:

x/<n><f><u> &<variable>	Examine memory location <variable>		
	<n>	number of locations to display, 1 is defualt.	
	<f>	format:	d – decimal
			x – hex
			u – unsigned
			c – character
			s – string
			f – floating point
	<u>	unit size:	b – byte (8-bits)
			h – halfword (16-bits)
			w – word (32-bits)
			g – giant (64-bits)

For example, to display the 16-bit variable **wVar2** and the 32-bit variable **dVar1**, the commands would be as follows:

```
x/dh &wVar2
x/dw &dVar1
x/dg &qVar1
```

For future assignments you will need to select the correct command to display the data based on the defined size and any guidance from the assignment.

Data Representation Worksheet

bVar1	**b**																			**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**				
	d																			**31**											
	h																			**1**				**F**							
bVar2	**b**																														
	d																														
	h																														
bAns1	**b**																														
	d																														
	h																														
bAns2	**b**																														
	d																														
	h																														
wVar1	**b**															**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**
	d															**3924**															
	h															**0**				**F**				**5**				**4**			
wVar2	**b**																														
	d																														
	h																														
wAns1	**b**																														
	d																														
	h																														
wAns2	**b**																														
	d																														
	h																														
dVar1	**b**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
	d	**174632105**																													
	h	**0**				**A**				**6**				**8**				**A**				**C**				**A**				**9**	
dVar2	**b**																														
	d																														
	h																														
dVar3	**b**																														
	d																														
	h																														
dAns1	**b**																														
	d																														
	h																														
dAns2	**b**																														
	d																														
	h																														

Data Representation Worksheet

flt1	**b**	1	1	0	0	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	d	-14.25_{10}																													
	h	C				1				6				4				0				0				0				0	
sign		- => 1																													
exp		10000010_2																													
mantisa		$11001000000000000000000_2$																													
flt2	**b**																														
	d																														
	h																														
sign																															
exp																															
mantisa																															
threePi	**b**																														
	d																														
	h																														
sign																															
exp																															
mantisa																															

	ltr 1	**ltr 2**	**ltr 3**	**ltr 4**	**ltr 5**	**ltr 6**
myClass =	**C**	**S**		**2**	**1**	**8**
hex	**43**	**53**	**20**	**32**	**31**	**38**
edName =						
hex						
myName =						
hex						