SI485H Stack Based Binary Exploits and Defenses
FALL 2015

NAME:_____

HW1

COLLABORATOR(S):_____

5/3/1/0 1. Label the different parts of the C program:

(a) number of command line arguments

(b) library function

(c) return value

(d) header files

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char * argv[]){
   printf("Hello World!\n");

   return 0;
}
```

2.

a) Explain the following compiler error:

```
user@si485H-base:demo$ gcc -o hello hello.c /tmp/ccC4VYbK.o:
In function `main': hello.c:(.text+0x20): undefined
reference to `world' collect2: error: ld returned 1 exit
status
```

5/3/1/0

b) What would the programmer need to provide in order to complete the compilation?

5/3/1/0

3. What is the difference between *compilation* and *assembling*?

5/3/1/0

4. **Circle** all the features that the OS provides:

5/3/1/0

(e) format printing

(f) writing to a terminal

(g) processing network packets

(h) executing a program

(a) Managing the file system

(b) String manipulation

(c) Managing allocated memory

(d) Allocating new memory

/25

NAME: _____

5. What is the difference between a system call and a library function?

5/3/1/0

6. Clone the following git repo to retrieve the binaries

git clone git@saddleback.academy.usna.edu:aviv/HW-1.git

Trace the following program using **ltrace** and **strace** and answer the questions below:

trace-me-1

a) What library functions are used in this program?

10/8/4/0

b) Given the strace, match those library functions to the associated system call, provide some explanation.

10/8/4/0

7. In the same cloned repo, you'll find the following program:

15/12/10/6/3/0

trace-me-2

Base on the trace, determine the *secret* of the program and how you did so:

(*hint read the strace output very carefully, and the man pages are your friend)*

/40

NAME: _____

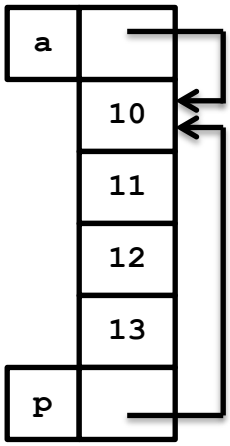
5/3/1/0 8. Convert the four byte hex values into their **signed** and **unsigned** values (*hint: use a computer)*:

		signed	**unsigned**
a)	0xffffffff	_____	_____
b)	0x0000000b	_____	_____
c)	0x8000000b	_____	_____
d)	0xdeadbeef	_____	_____

9. Complete the memory model for the following sample program at each of the labeled marks: (Note, use arrows for pointers)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char * argv[]){

  int a[] = {10,11,12,13};
  int * p = a; //MARK 0

  p++; //MARK 1

  p[1] = 50;

  p--; //MARK 2

  *p = 12; //MARK 3

}
```

MARK 0



5/3/1/0 a) MARK 1

5/3/1/0 b) MARK 2

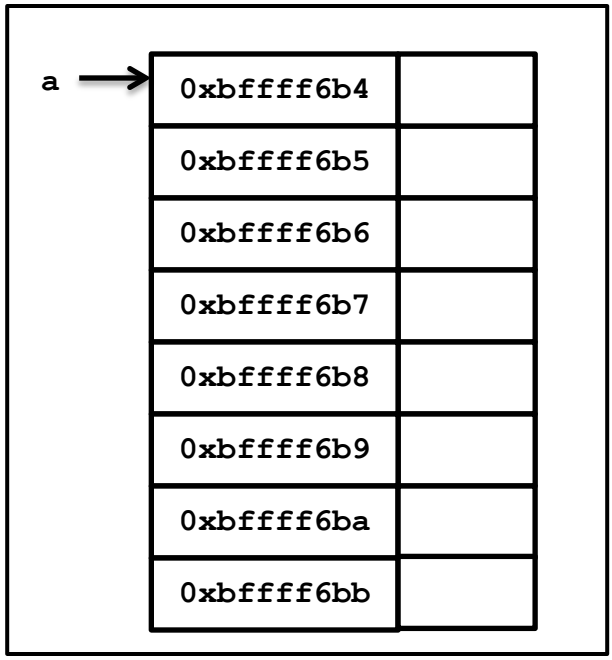
5/3/1/0 c) MARK 3

/20

NAME: _____

10/8/4/0 10. Fill in the memory diagram for array values that is byte by byte. (*hint: note the memory addresses in the memory diagram)*

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char * argv[]){

  int a[] = {0xcafebabe,0xdeadbeef};
  char * p = (char *) a;

  p++;
  *p = 0x00; //MARK
}
```



11. In the below program, match the section of the program memory layout to where that variable's value is stored.

```
#include <stdio.h>
#include <stdlib.h>

int mystrlen(int *s){
  int i;
  for(i=0;*s++;i++);
  return i;
}

int main(int argc, char * argv[]){

  char a[] = "hello";
  char * b = getenv("PATH");
  char * c = "world";
}
```

5/3/1/0

a) &a _____

b) b _____

c) c _____

d) mystrlen _____

- Reserved
- stack
- heap
- text
- bss

/15