

0. Fundamentals:

Choose and answer 4 of the following 7 parts. Be sure to **indicate which 4** you want graded.

- a. What are the differences between unions and structs?
- b. What command would you use to stop a process one of your programs fork()ed?
- c. If a file had the permissions '-wr--r-w-x' write out what capabilities each user type has?
- d. What two subdirectories are in every single directory?
- e. How is the 'top' shell command different from 'ps'?
- f. What is the difference in size between an int pointer and a short pointer? (Presuming an int is 4 bytes and a short is 2 bytes)
- g. What function do you call once that can return twice?

1. Error Check: Oops...

Answer **all** of the below.

Find, circle and explain the error(s). Write the correction to the right of the error(s).

a. This code is supposed to increment the integer pointed to by its parameter, but breaks...why?

```
int * increment(int * num)
{
    int newval = *num;
    newval++;
    int * retval = &newval;

    return retval;
}
```

b. This code is supposed to read in a command of length, maximum, but often segfaults...why? (presume the commands given are always correct)

```
int readInCommand()
{
    char * buffer = (char*)malloc(8*sizeof(char));
    int i = 0;
    int status = 0;

    do
    {
        status = read(STDIN, buffer+i,1);
        ++i;
    }
    while(status > 0 && i < 8);

    if(status < 0)
    {
        return status;
    }
    else
    {
        return strlen(buffer);
    }
}
```

2. Type of types

Answer **all** of the following.

Some may not be answerable - if so, mark them as such and explain why.

Presume:

chars are 1 byte, shorts are 2 bytes, ints are 4 bytes, doubles are 8 bytes, pointers are 8 bytes

a. Given the following code, what value does mystery hold? Explain why.

```
uncertain * item = (uncertain*)malloc(8*sizeof(uncertain));
int mystery = 0;
mystery = (item + 1) - item;
```

b. Will the following code work or segfault? Explain why.

```
char * buffer = (char*)malloc(sizeof(char*));
*buffer = "login>";
printf("%s\n", *buffer);
```

c. What are minimum and maximum sizes of 'stuff' and 'things' below? Explain why.

union stuff	struct things
{	{
int a;	int a;
char b;	char b;
double c;	double c;
}	}

d. Based on the code below, determine the address of the pointer after the addition.

```
printf("addr: %d\n", ptr); // → outputs: 9126123
```

```
ptr = ptr + 2;
```

```
printf("addr: %d\n", &ptr); // → outputs: _____
```

3. What am I? Pointer charade!

Answer **all** of the following.

Based on the code below identify what each of the expressions below is; an address, integer, character, or likely to segfault. If the value is known, write the value as well.

Presume:

the code compiled correctly despite warnings

chars are 1 byte, shorts are 2 bytes, ints are 4 bytes, doubles are 4 bytes

Let:

```
int x = 27;
```

```
char *y = (char*)malloc(sizeof(int));
```

```
*((int*)y) = x;
```

```
char c[8] = "hithere";
```

```
char *ptr = (char*)malloc(sizeof(short)*4);
```

```
strcpy(ptr,c);
```

```
char ** morePtr = (char*)malloc(sizeof(char*)*3);
```

```
*morePtr = ptr;
```

a. *morePtr

f. *(&x)

b. *ptr

g. morePtr[0][1]

c. &y

h. *((morePtr)+1)

d. x

i. *(c+3)

e. y[2]

j. *x

4. SysOps

Answer ***all*** of the following.

a. Why does `fork()` return different values to the child and parent Process?

b. What does `exec()` do?

c. What does a non-blocking call do differently than a blocking call?

d. How does a C signal handler differ from a Java Exception?