Sample Final Answers

1. Here is a macro for computing the maximum of two munbers:

```
#define max(x, y)((x) > (y) ? (x) : (y))
```

and here is a function:

```
int max(int a, int b)
{
      return(a > b ? a : b);
}
```

In the following code fragment, give the values of a, b, c, d, and e at the end if max is the above macro, and then if max is a function call:

```
a = -1;
b = 0;
c = 1;
d = max(++a, b);
e = max(c++, b);
```

variable	value if max is macro	value if max is function
a		
b		
c		
d		
e		

Answer:

variable	value if max is macro	value if max is function
a	0	0
b	0	0
c	3	2
d	0	0
e	2	1

2. Here is a very simple program which is designed to sum its arguments:

```
#include <stdio.h>

int main(int argc, char *argv[])
{
      int sum = 0;
      int i;

      for(i = 1; i < argc; i++)
            sum += argv[i];
```

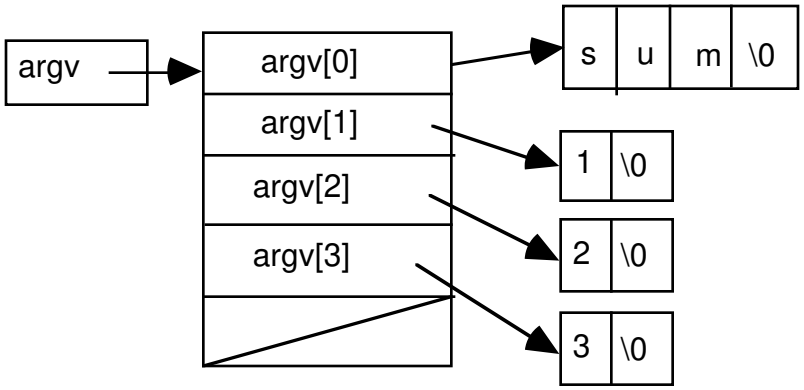
```
        printf("%d\n", sum);

        return(0);
}
```

a. Suppose this program is invoked as

```
sum 1 2 3
```

What are the values of argc and the elements of argv? Draw a picture if that would help you indicate what the values are. Please note I want the value of argc, argv, and each element of argv, and so forth, either as a number or character, or as arrows in a picture (for addresses).

Answer: argc is 4, and the following summarizes argv:



b. When I ran this programs, I got 2147431340. Why does the program print the incorrect value, and how would you fix it? (If you cannot write the code, for partial credit simply say what you would do.)

Answer: The program adds in the addresses of the strings, not the numbers those strings represent. The easiest way to fix the program is to replace the line

```
sum += argv[i]
```

with the line

```
sum += atol(argv[i])
```

3.

a. What does the following function do? Remember, its purpose can be stated in a succinct sentence; for partial credit, you can describe what each line does, but for full credit, you must state its function in one very short sentence.

```
int mystery(char a[])
{
        register int i;

        for(i = 0; a[i]; i++)
                if (!isdigit(a[i]))
                        return(0);
        return(1);
}
```

Answer: It returns 1 if the array a[] contains all digits; that is, it is an unsigned number. Otherwise, it returns 0.

b. Rewrite the function using pointers rather than an array.

Answer:

```
int mystery(char *a)
{
```

```
        register int i;

        for( ; *a; a++)
              if (!isdigit(*a))
                    return(0);
        return(1);
}
```

4. If the current working directory is /usr/bishop/tmp, give an absolute path name without any . or .. directories in it for:

   a. ../src/./../passwd+/misc/./root.c

      Answer: The absolute path name, with . and .. directories, is /usr/bishop/tmp/../src/./../passwd+/misc/./root.c so, removing the . and .. directories, we have /usr/bishop/passwd+/misc/root.c

   b. /usr/holly/files/../DMS/./../DMS/schedule

      Answer: This is an absolute path name; removing the . and .. directories, we have /usr/holly/DMS/schedule

5. Rewrite the following command without using temporary files

```
tr A-Z a-z < /usr/dict/words > /tmp/X
grep '^bananna$' /tmp/X > /tmp/Y
wc -l /tmp/Y
```

   Answer: Use pipes:

```
tr A-Z a-z < /usr/dict/words | grep '^bananna$' | wc -l
```

6. What is the value of n after the following code executes?

```
char charray[] = "A-DF123";
int n, s = 1;
char *p;

for(n = 0, p = charray; *p; p++){
      switch(*p){
      case '-':
            s = -s;
            break;
      case '0':
      case '1':
      case '2':
      case '3':
      case '4':
      case '5':
      case '6':
      case '7':
      case '8':
      case '9':
            n = n * 10 + (*p - '0');
            break;
      }
}
n = s * n;
```

   Answer: n = –123.