Question 1. [6 MARKS]

All parts of this question assume the following C statement. Parts (b) through (e) assume a variable called `ptrs`.

```
char data[128] = "How much wood could a woodchuck chuck?";
```

Part (a) [1 MARK] Is the following statement allowed? Why or why not?

```
data[10] = 'f';
```

Yes, because data is an array that is statically initialized by the string literal. Even though we can't modify a string literal, we can modify and array of characters.

Part (b) [1 MARK]

We want to use a variable `ptrs` to point to the first character of each word in `data`. Fill in the argument to `malloc` so that enough memory is allocated to store a pointer to the first character of each word in `data`.

```
char **ptrs = malloc(7 * sizeof(char*));
```

Part (c) [1 MARK]

Using array notation, write a statement that makes the first element in `ptrs` refer to the first character of the first word in `data`.

```
ptrs[0] = &data[0];
```

Part (d) [1 MARK]

Using only pointer notation, write a statement that makes the second element in `ptrs` refer to the first character of the second word in data without changing the value of `ptrs`.

```
*(ptrs+1) = data + 4;
```

Part (e) [2 MARKS] Write two C statements so that the following statement prints "wood".

```
ptrs[0] = &data[9];
data[13] = '\0';



printf("%s\n", *ptrs);
```

Question 2. [4 MARKS]

The current working directory contains two files shown below with their contents.

cmd1	cmd2
wc -w	*1

Recall that `wc -w FILE` prints the number of words in FILE followed by name of FILE.

Two variables are defined below. For each of the following lines, give the output that would appear when the command is run. (` is a backquote and ' is a forward quote)

```
set cat *


echo '$*'  =>      $*


echo "$*"  =>      cat cmd1 cmd2


echo `$*`  =>      wc -w cmd1


x=`$1 $3`
echo $x    =>        cmd1
```

Question 3. [8 MARKS]

Write a shell program that makes a list of all the define variables in the `.h` files in the current working directory. It then prints the name of each variable followed by the number of lines that the variable appears in in all of the C source files in the current working directory.

For example if a header file contained the line

```
#define MAXFILES 100
```

and the variable `MAXFILES` appeared on one line in `a.c` and twice in `b.c` then one line of output would be

```
MAXFILES 3
```

Tip: When the output of a command is assigned to a variable, newline characters are removed to form a single list.

```
for f in *.h
do
    defines=`grep define $f | cut -d " " -f 2`
    for d in $defines
    do
        echo $d `grep $d *.c| wc -l`
    done
done
```

Question 4. [9 marks]

Part (a) [8 marks]

Complete the C function below that returns a new string `line` where the first occurrence of the string `oldvalue` has been replaced with the string `newvalue`. If `oldvalue` does not occur in `line` then return the original line.

You are permitted to alter `line`. You must not use `strcpy` or `strcat`, but you may use `strncpy` and/or `strncat`. For full marks the arguments to either must be correct. You may assume that the resulting line will never be larger than 256 characters.

For example:

```
char sentence[256] = "The cow jumped over the moon.";
char *p = replace("cow", "horse", sentence);
printf("%s\n", p);
// prints "The horse jumped over the moon.";
```

```
    char *replace(char *oldvalue, char *newvalue, char *line) {


        char *result = malloc(256);
        char *ptr = strstr(line, oldvalue);
        if(ptr != NULL) {
            return line;
        }

        *ptr = '\0';
        strncpy(result, line, 256);

        ptr++;
        int i = 0;
        while(oldvalue[i] == *ptr) {
            i++;
            ptr++;
        }
        strncat(result, newvalue, 256 - strlen(result));
        strncat(result, ptr, 256 - strlen(result));
        return result;
    }
```

Part (b) [1 mark] If the function above did not return a pointer to the resulting string, could we set `line` to point to the new string so that when the function returns the results would still be visible? Explain briefly.