Homework #2

Due: Thursday, October 16, 1996, at 6:00 PM
Points: 130

The UNIX Environment

Submitting: Type your answers into a file called *Answers*, and submit it as described in the **All About Homework** handout. Remember, we'll read it on the DECs, so be sure it can be read on them! (**Note**: if you use another file name, we will deduct 5 points.)

The purpose of these questions is to have you use the UNIX system and environment.

1. (*10 points*) What is the parent directory of /?
2. (*10 points*) What is the command that executes with process id (PID) 2?
3. (*10 points*) How would I make an alias that will print the name of the current working directory whenever I execute the C-Shell command *cd*? (Assumeyou are using the C-Shell.)

The C Programming Language

Submitting: Put your program into a file called *main.c*, and submit it as described in the **All About Homework** handout. Remember, we'llcompile and run it on the DECs, so be sure it works there! (**Note**: if you use another file name, we will deduct 5 points.)

(*100* points) The purpose of this question is for you to build on your (or my) answer to program 1 to gain experience with poiinters and dynamic storage allocation..

Input Specifications

Read from the standard input. The input is supposed to be a C program. You need not check that it is, in fact, a legal C program; your program should simply assume it is.

Output Specifications

Write to the standard output. In the first 6 columns of each output line, print a line number followed by a period ".". Print the input line beginning in column 9 (one tab from the left margin)..

If the line is a preprocessor control line, and the preprocessor directive is **define**, add the name, line number on which the **define** occurs, and definition to a table. When the input is finished, print the table, headed by the title "Table of Defines" and beginning on a separate page.

Other Constraints

Put the entries into an array, one definition and one name per entry. (So, you will have to use a structure for this.) The form of each entry is

```
struct defent {
        char *name;
        char *defn;
};
```

You must allocate space for the `name` and `defn` fields, but you may assume that no more than 1000 symbols will be defined.

Example Input

```
#define PI 3.14159
#define E 2.71828

int main(void)
{
      printf("pi = %f, e = %f\n", PI, E);
}
```

Example Output

```
    1.  #define PI 3.14159
    2.  #define E 2.71828
    3.
    4.  int main(void)
    5.  {
    6.      printf("pi = %f, e = %f\n", PI, E);
    7.  }

Table of Defines:
name            line                    definition
PI                1                   3.14159
E                 2                   2.71828
```

Extra Credit

Submitting: Create a C program called *extra.c*. You may copy your *main.c* into it and modify that (in fact, we encourage you to!)

Modify your answer so that there can be anyt number of symbols, not simply 1000..