



Program 1 Write-Up

Systems Programming

Summa

| | |
|---------------------|---|
| 1 - Program 1 | 1 |
|---------------------|---|

On Honeybee, the following code was written (after MUCH debugging):

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

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

    char expandfrom;
    char expandto;
    int tabStop;
    for (int i = 1; i < argc; i++){
        if (argv[i][0]=='-'){
            switch (argv[i][1]){
                case 't':
                    expandfrom = '\t';
                    expandto = ' ';
                    break;
                //putting this here for future expansion of this program... probably wont work
                //well now
                case 's':
                    expandfrom = ' ';
                    expandto = '\t';
                    break;
                default :
                    fprintf(stderr, "The character %s is not a valid expansion type\n", argv[i]);
            }
        } else {
            if (argc > 2){
                expandfrom = '\t';
                expandto = ' ';
            }
            tabStop = atoi(argv[i]);
        }
    }

    char curr;
    int column = 1;
    while((curr = getchar())!= EOF){
        //check if current char is the character to be expanded from
        if (curr == expandfrom){
            //for some reason I have to deal with the case when the current column is an
            //integer multiple of the tabstop
            if (column % tabStop == 0){
                putchar(expandto);
                column ++;
            } else {
                //for the specified operation, replace the tab with enough spaces to get to the
                //next tab stop
                for (int i = 0; i < tabStop + 1 - column % tabStop; i++){

                    putchar(expandto);
                }
            }
        }
    }
}
```

}

1

1

For the command **myexpand 3 <tabtest.txt**, the following results were obtained:

```

nrlibby@honeybee:~/cos350/prog$ myexpand 3 < tabtest.txt
tab test for prog1 in COS350
0
1      1
22     2
333    3
4444   4
55555  5
666666 6
7777777 7
88888888 8
1      1      1      1      1
22     22     22     22     22
333    333    333    333    333

#####          # #####
#           # #           # #           #
#           # #           # #           #
#       ## # # #       # #       # #       #
#           # #       # #           # #           #
#           # #       # #           # #           #
#####  #####  #####  #####  #####

```

Clearly, my console input processing needs work, but I think its a minor flaw in the for loop over all, as I have debugged and found that for case 2 and 3, it simply never reaches the definition of **tabStop**.