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

char comm[100], *ptr, *args[10];
int tot;

int list(char* option, char* dirname);

main()
{
        do
        {
                printf("myshell$ ");
                getcomm();
                sep_arg();
                take_action();
                printf("\n");
        } while(1);
}

sep_arg()
{
        char* token;
        tot = 0;

        token = strtok(comm, " ");

        while(token != NULL)
        {
                args[tot] = (char*)malloc(20);
                strcpy(args[tot++], token);
                token = strtok(NULL, " ");
        }
        return;
}

getcomm()
{
        int len;

        ptr = fgets(comm, 80, stdin);
        len = strlen(comm);
        comm[len-1] = '\0';
        return;
}

take_action()
{
```

```
        if(strlen(comm) <= 1)
                return;
        else if(strcmp(args[0],"list") == 0)
                list(args[1], args[2]);
        else
        {
                printf("Command not found!");
                exit(0);
        }
}

int list(char *option, char *dirname)
{
    DIR *dp;
    struct dirent *dent;
    int cnt = 0;

    dp = opendir(dirname);
    if (dp == NULL)
    {
        printf("\nUnable to open directory\n");
        return;
    }
    dent = readdir(dp);


    if (strcmp(option, "f") == 0)
    {
        while (dent != NULL)
        {
            printf("\n%s", dent->d_name);
            dent = readdir(dp);
        }
    }
    else if (strcmp(option, "n") == 0)
    {
        while (dent != NULL)
        {
            cnt++;
            dent = readdir(dp);
        }
        printf("\ntot dir entries = %d", cnt);
    }
    else if (strcmp(option, "i") == 0)
    {
        while (dent != NULL)
        {
            printf("\n%s %ld", dent->d_name, dent->d_ino);
            dent = readdir(dp);
        }
```

```
    }
    else
        printf("\nInvalid option");

    printf("\n");
    closedir(dp);
}
```