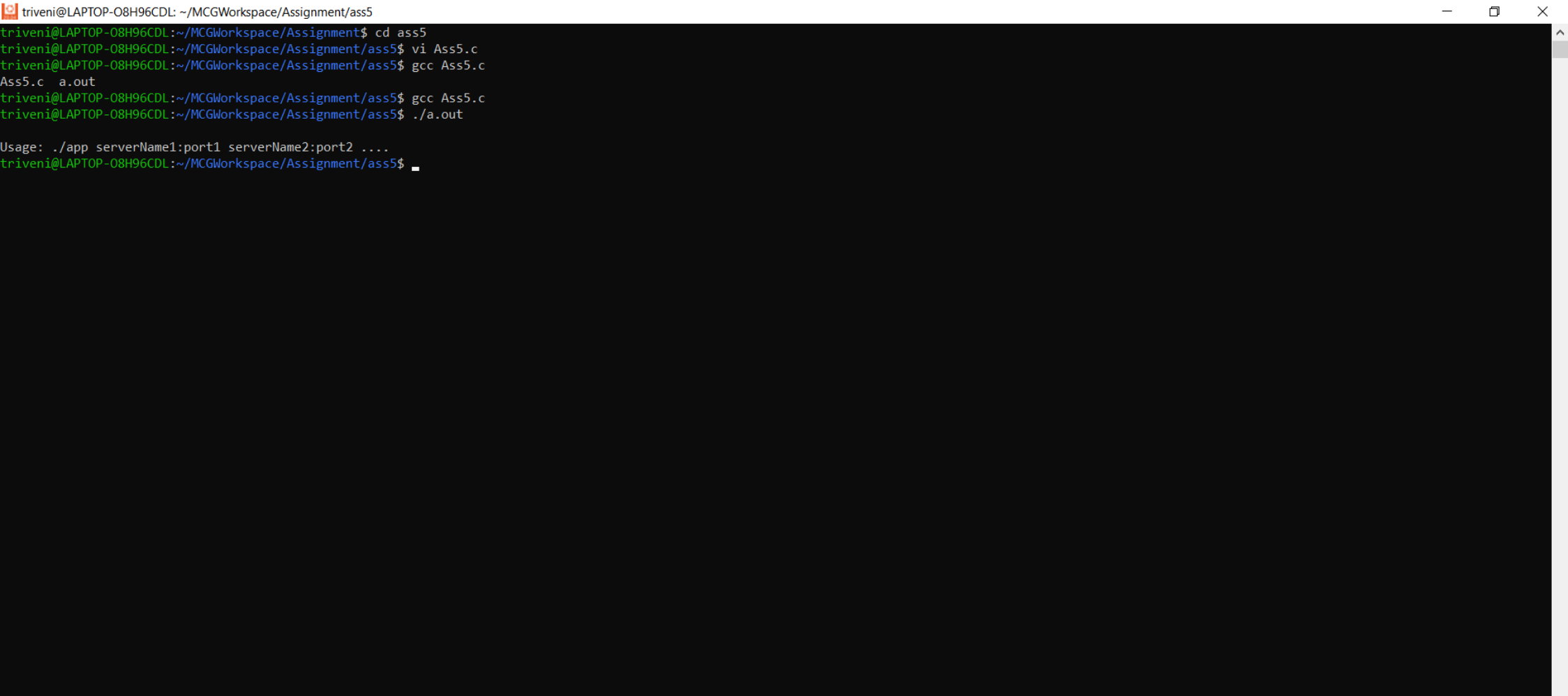
triveni@LAPTOP-O8H96CDL: ~/MCGWorkspace/Assignment/ass5
triveni@LAPTOP-O8H96CDL:~/MCGWorkspace/Assignment$ cd ass5
triveni@LAPTOP-O8H96CDL:~/MCGWorkspace/Assignment/ass5$ vi Ass5.c
triveni@LAPTOP-O8H96CDL:~/MCGWorkspace/Assignment/ass5$ gcc Ass5.c
Ass5.c a.out
triveni@LAPTOP-O8H96CDL:~/MCGWorkspace/Assignment/ass5$ gcc Ass5.c
triveni@LAPTOP-O8H96CDL:~/MCGWorkspace/Assignment/ass5$ ./a.out
Usage: ./app serverName1:port1 serverName2:port2 ....
triveni@LAPTOP-O8H96CDL:~/MCGWorkspace/Assignment/ass5$

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

int getWords(char *, char*, char *);
int validSName(char *);
int validPNum();
void concatSName(char *, char *, int );

int main(int argc, char *argv[])
{
        int i, cwords=0, retValue=0;
        char servNames[10][100] = { '\0',};
        char portNums[10][10] = {'\0',};
        char cSNames[100] = {'\0', };
        if(argc <= 1)
        {
                printf("\nUsage: ./app serverName1:port1 serverName2:port2 ....\n");
                exit(EXIT_FAILURE);
        }
        for(i=1, cwords=0;i<argc;i++,cwords++)
        {
               retValue = getWords(servNames[cwords], portNums[cwords], argv[i]);
               if(retValue == 1)
                       cwords--;
        }

        for(i=0;i<cwords;i++){
               printf("\n%s",servNames[i]);
               if(i==cwords-1)
               {
                       concatSName(servNames[i], cSNames, 0);
               }
               else
                       concatSName(servNames[i], cSNames, 1);
               }
               printf("\n%s",cSNames);
               printf("\n\n");
               return 0;
}
int getWords(char *sNames, char *pNums, char *argv)
{
               char *tokens;
               tokens = strtok(argv, ":");/*printf("\n%s",tokens);*/
               if(validSName(tokens) == 0)
               {
                       return 1;/*printf("\nNot a valid server name\n");*/
               }
```

```
                }
                strcpy(sNames,tokens);
                tokens = strtok(NULL, ":");
                if(validPNum(tokens) == 0)
                {
                        return 1;/*printf("\nNot a valid port number\n");*/
                }
                /*printf("\n%s",tokens);*/
                if((atoi(tokens)>=0)&&(atoi(tokens)<=1023))
                        strcpy(pNums, tokens);
                else
                {
                        return 1; /*printf("\n port number out of range\n");*/
                }
                return 0;
}

int validSName(char *sName)
{
                int size = 0, i;
                int flag = 0;
                if(sName != NULL){
                        size = strlen(sName);
                        for(i=0;i<size;i++)
                        {
                                /*printf("%c",sName[i]);*/
                                if(isalpha(sName[i])== 0)
                                {
                                        flag = 1;
                                        break;
                                }
                        }
                        /*printf("\n")*/
        }
        else
                flag = 1;
        if(flag == 1)
                return 0;
        return 1;

}
int validPNum(char *pNum)
{
        int size = 0, i;
        int flag = 0;
        if(pNum != NULL){
                size = strlen(pNum);
                for(i=0;i<size;i++)
                {
```

```
                        {
                                /*printf("%c",sName[i]);*/
                                if(isalpha(sName[i])== 0)
                                {
                                        flag = 1;
                                        break;
                                }
                        }
                        /*printf("\n")*/
        }
        else
                flag = 1;
        if(flag == 1)
                return 0;
        return 1;

}
int validPNum(char *pNum)
{
        int size = 0, i;
        int flag = 0;
        if(pNum != NULL){
                size = strlen(pNum);
                for(i=0;i<size;i++)
                {
                        if(isdigit(pNum[i])== 0)
                        {
                                flag = 1;
                                break;
                        }
                }
        }
        else
                flag = 1;


        if(flag == 1)
                return 0;
        return 1;
}
void concatSName(char *sNames, char *cSNames, int n)
{
        int i;
        strcat(cSNames, sNames);
        if(n == 1)
                strcat(cSNames,"_");
}
```