WEEK2 _CD

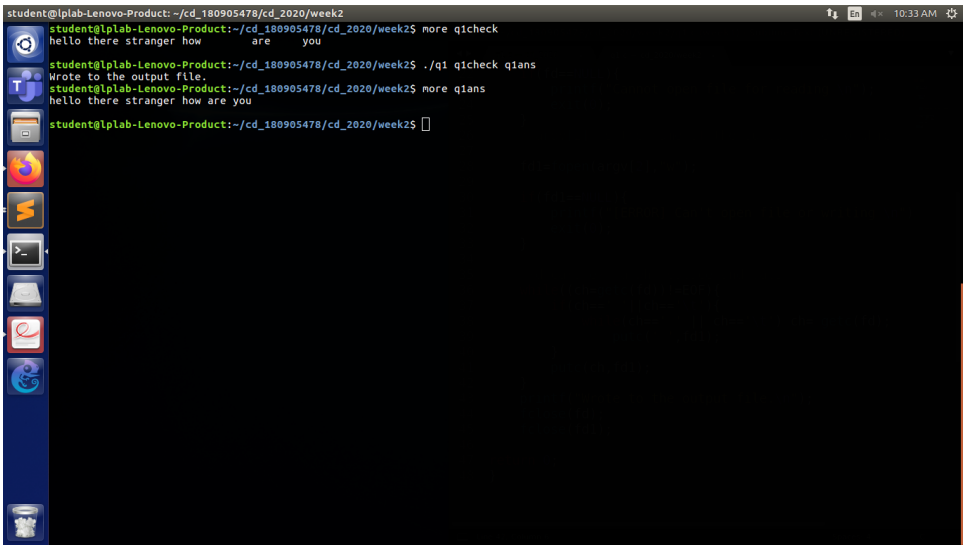
1. Program that takes file as input and replaces blank spaces and tabs by single space and writes the output to a file.

   Code:

```
/*
AUTHOR: SAGNIK CHATTERJEE
DATE : 9 DEC,2020
USAGE: ./q1 file1 file2
file1 is the input file
file2 is the output file

*/

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


int main(int argc ,char **argv){
   FILE *fd , *fd1;
   char buffer[1024],ch ;

   //argv[1] is the name of the input file passed
   fd=fopen(argv[1],"r");

   if(fd==NULL){
      printf("Cannot open file for reading \n");
      exit(0);
```

```
    }
    //argv[2] is the name of the output file to which result is written

    fd1=fopen(argv[2],"w");

    if(fd1==NULL){
        printf("[ERROR] Can't open file or writing \n");
        exit(0);
    }

    //dicard extra whitespace and tabs
while((ch=getc(fd))!=EOF){
        if(ch==' '||ch=='\t'){
            while(ch==' ' || ch=='\t') ch= getc(fd);
                putc(' ',fd1);
        }
        putc(ch,fd1);
    }
  printf("Wrote to the output file.\n");
    fclose(fd);
    fclose(fd1);

return 0;
}
```

Scrreshot :



2. Program to discard processor directives from the given input c file.

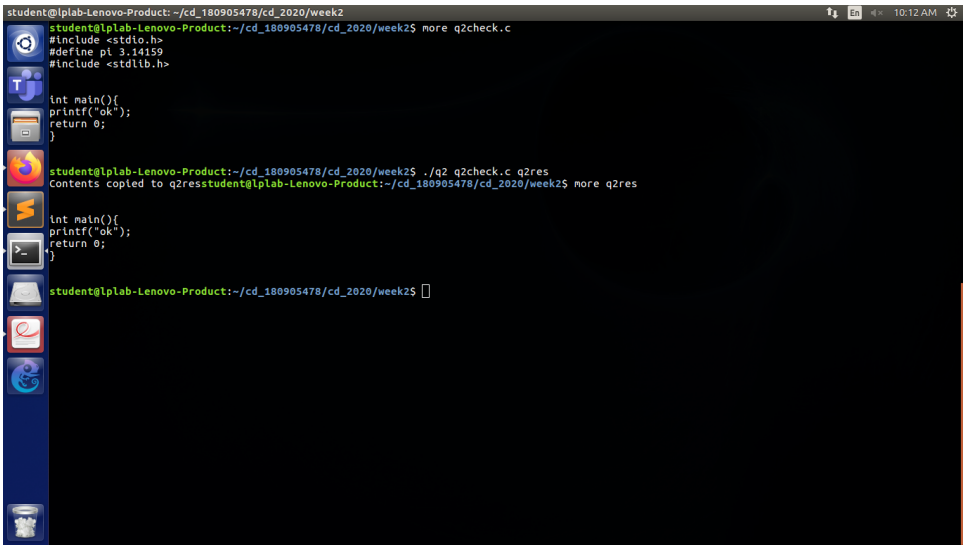
CODE:

```
/*
AUTHOR :SAGNIK CHATTERJEE
DATE :dec9,2020
Usage : ./q2 file1  file2

file1 is the input C file
file2 is the result file

Here input taken from file1check.c and written to file2
*/

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

```
int main(int argc,char **argv){
   FILE *fa, *fb;
   //argv[1] is the input file for reading

   char ch;
   fa=fopen(argv[1],"r");
   fb=fopen(argv[2],"w");
   if(fa==NULL){
      printf("[ERROR] Couldnt the open for reading .\n");
      exit(0);
   }
   if(fb==NULL){
      printf("[ERROR] Couldn't open the file for writing.\n");
      exit(0);
   }
   char buffer[1024];//the buffer from whcih we will write to fd
   char ignore[1024];//we would read the line but will write to this ignore array

   char *line=NULL;
   size_t len=0;
   ssize_t read;
   buffer[1023]='\n';
   while((read =getline(&line,&len,fa))!=-1){
      if(strstr(line,"#define")!=NULL || strstr(line,"#include")!=NULL){
      continue;
      }
      else{
         fputs(line,fb);
      }
```

```
    }


    printf("Contents copied to %s",argv[2]);
    fclose(fa);
    fclose(fb);
    return 0;
}
```

Screenshot:



3. Program takes c file as input and recognises all the keywords and prints them in upper case.

CODE:

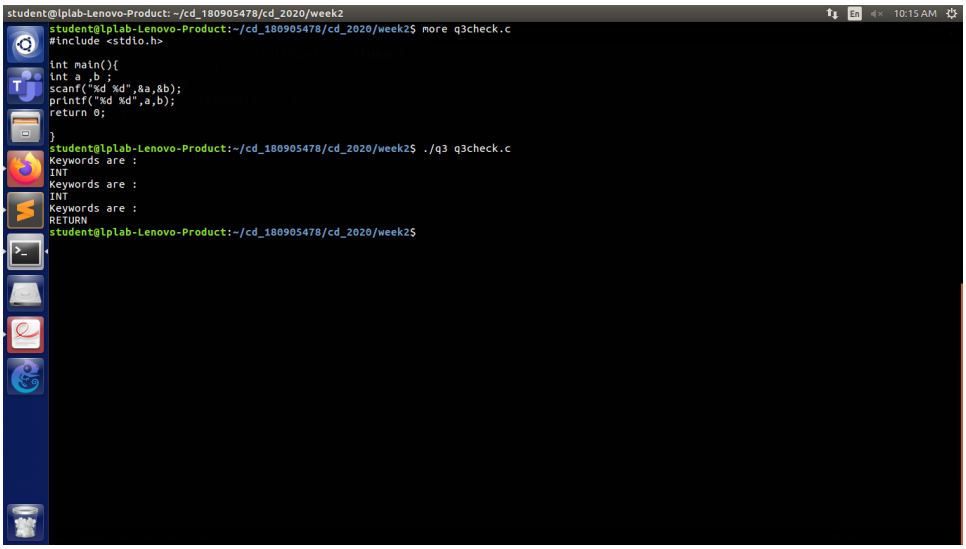
```
/*
AUTHOR: SAGNIK CHATTERJEE
DATE : DEC 9,2020
USAGE : ./q3 file1
where file1 is the input c file


prints the result to the console
```

```
*/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
//#include <stdbool.h>
#include <stddef.h>

const char *keywords[32] = {
   "auto","double","int","struct","break","else","long",
   "switch","case","enum","register","typedef","char",
   "extern","return","union","continue","for","signed",
   "void","do","if","static","while","default",
   "goto","sizeof","volatile","const","float","short",
   "unsigned"
};

const char delimiters[]=" .,;:!-()\n\t";

int isKeyword (char *word) {
   int i;
   for (i = 0; i < 32; ++i) {
      if (strcmp(word, keywords[i]) == 0) {
         return 1;
      }
   }
   return 0;
}
```

```
void printUpperCase (char *word) {
   int l = strlen(word);
   char z;
   int i;
   printf("Keywords are :\n");
   for (i = 0; i < l; ++i) {
      z = word[i];
      printf("%c", z > 96 ? z - 32 : z);
   }
   printf("\n");
}

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



   FILE *fd1;


   //argv[1] is the input c file we give
   fd1=fopen(argv[1],"r");


   if(fd1==NULL){
      printf("[ERROR] Can't open the file to read from. \n");
      exit(0);
   }
   char buffer[1024];
   while(fgets(buffer,1024,fd1) >0){
      //temp copy of string
```

```
        char *cp =(char*)malloc(1024*sizeof(char));
        strcpy(cp,buffer);

        char *token=(char*)malloc(256*sizeof(char));

        do {//strsep :- extract the token from string,returns null if token not found
            token =strsep(&cp,delimiters);
            if(token!=NULL)
            {
                if(isKeyword(token)){
                    printUpperCase(token);
                }
            }
        }while(token!=NULL);
    }

    fclose(fd1);
    //while writing to file use toupper() to make it capital
    //since toupper() runs on a char , run a loop or can use ascii style
return 0;
}
```

SCREENSHOT:


student@lplab-Lenovo-Product: ~/cd_180905478/cd_2020/week2
10:15 AM
student@lplab-Lenovo-Product:~/cd_180905478/cd_2020/week2$ more q3check.c
#include <stdio.h>
int main(){
int a ,b ;
scanf("%d %d",&a,&b);
printf("%d %d",a,b);
return 0;
}
student@lplab-Lenovo-Product:~/cd_180905478/cd_2020/week2$ ./q3 q3check.c
Keywords are :
INT
Keywords are :
INT
Keywords are :
RETURN
student@lplab-Lenovo-Product:~/cd_180905478/cd_2020/week2$