**COMPILER DESIGN LAB 2**

**Q1. That takes a file as input and replaces blank spaces and tabs by single space and**

**writes the output to a file.**

**Program:**

#include <stdio.h>

#include <stdlib.h>

int main()

{

FILE \*fa, \*fb;

int ca, cb; int bs=' ';

fa=fopen("sample1.txt", "r");

fb=fopen("sample1out.txt","w");

if(fa==NULL || fb == NULL)

{

printf("Invalid files\n");

exit(0);

}

ca=getc(fa);

while(ca!=EOF)

{

if(ca!='\t' && ca!=' ')

{

putc(ca,fb);

}

else

{

if(ca=='\t')

putc(bs,fb);

else

{

while(ca==' ')

ca=getc(fa);

putc(bs,fb);

putc(ca,fb);

}

}

ca=getc(fa);

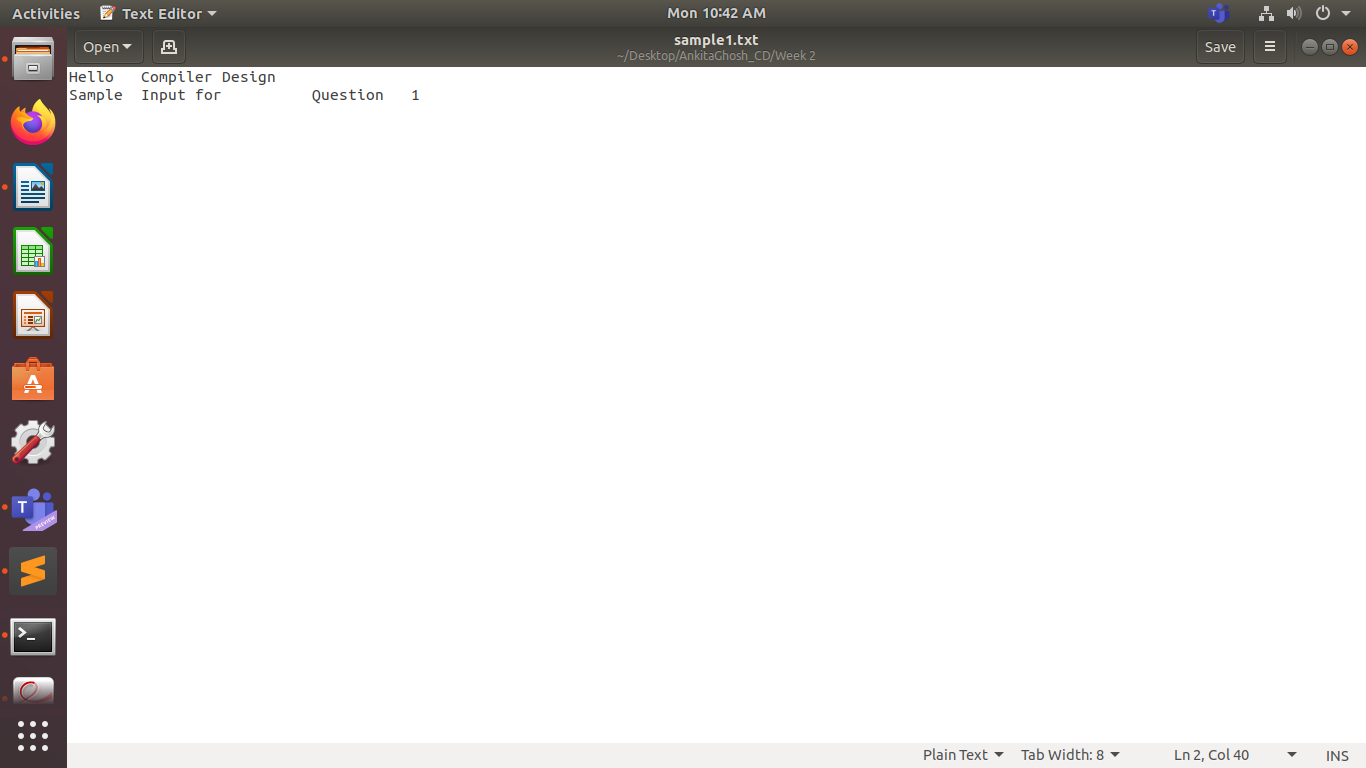
}

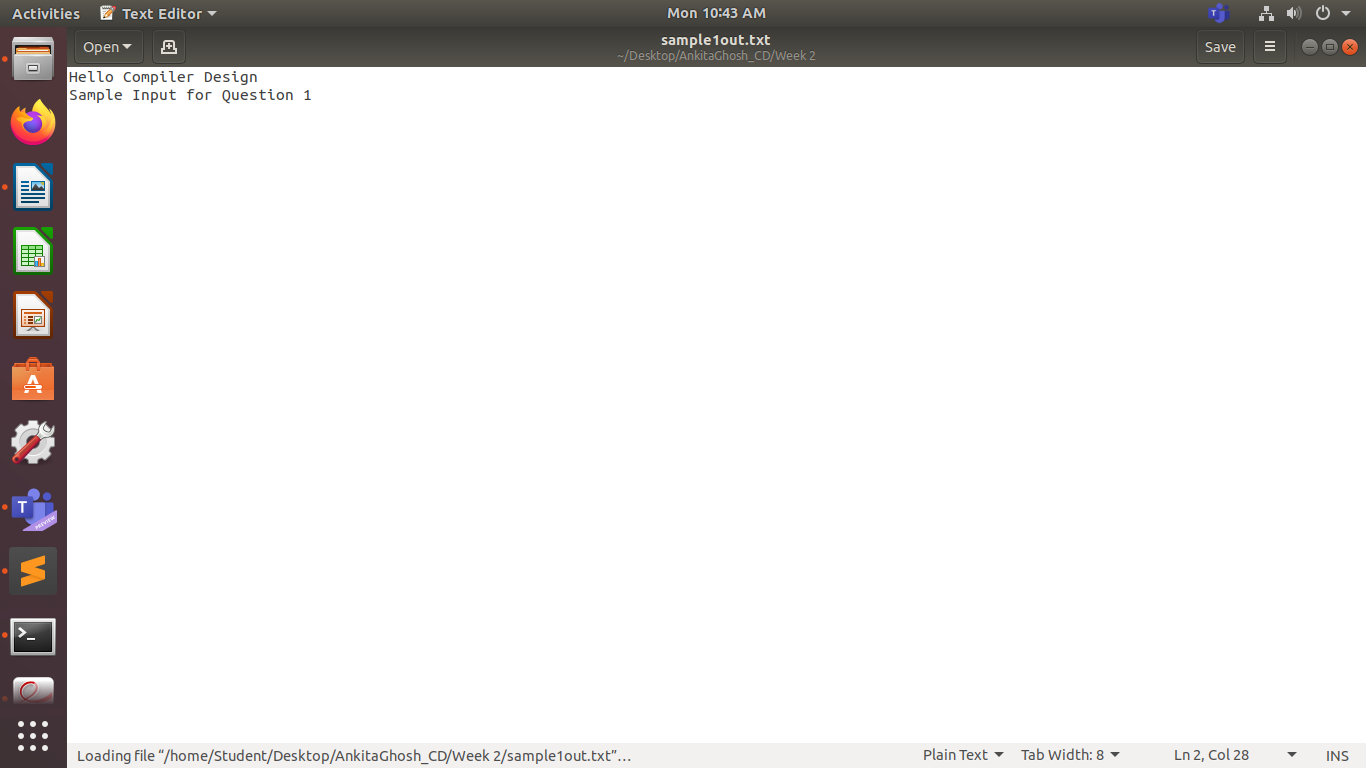
fclose(fa);

fclose(fb);

}

**Output:**





**Q2. To discard preprocessor directives from the given input ‘C’ file.**

**Program:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

const char \*directives[] = {"#include","#define","#if"};

int is\_directive(const char \*str)

{

for(int i = 0; i < sizeof(directives)/sizeof(char \*); i++)

{

int len = strlen(directives[i]);

if(strncmp(str, directives[i], len) == 0)

{

return 1;

}

}

return 0;

}

int main()

{

char buf[2048];

FILE \*f1,\*f2;

f1 = fopen("input.c", "r");

f2 = fopen("output2.c", "w");

if(f1 == NULL || f2 == NULL)

{

perror("Invalid files\n");

return 1;

}

while(fgets(buf, 2048, f1) != NULL)

{

if(!is\_directive(buf))

{

fputs(buf, f2);

}

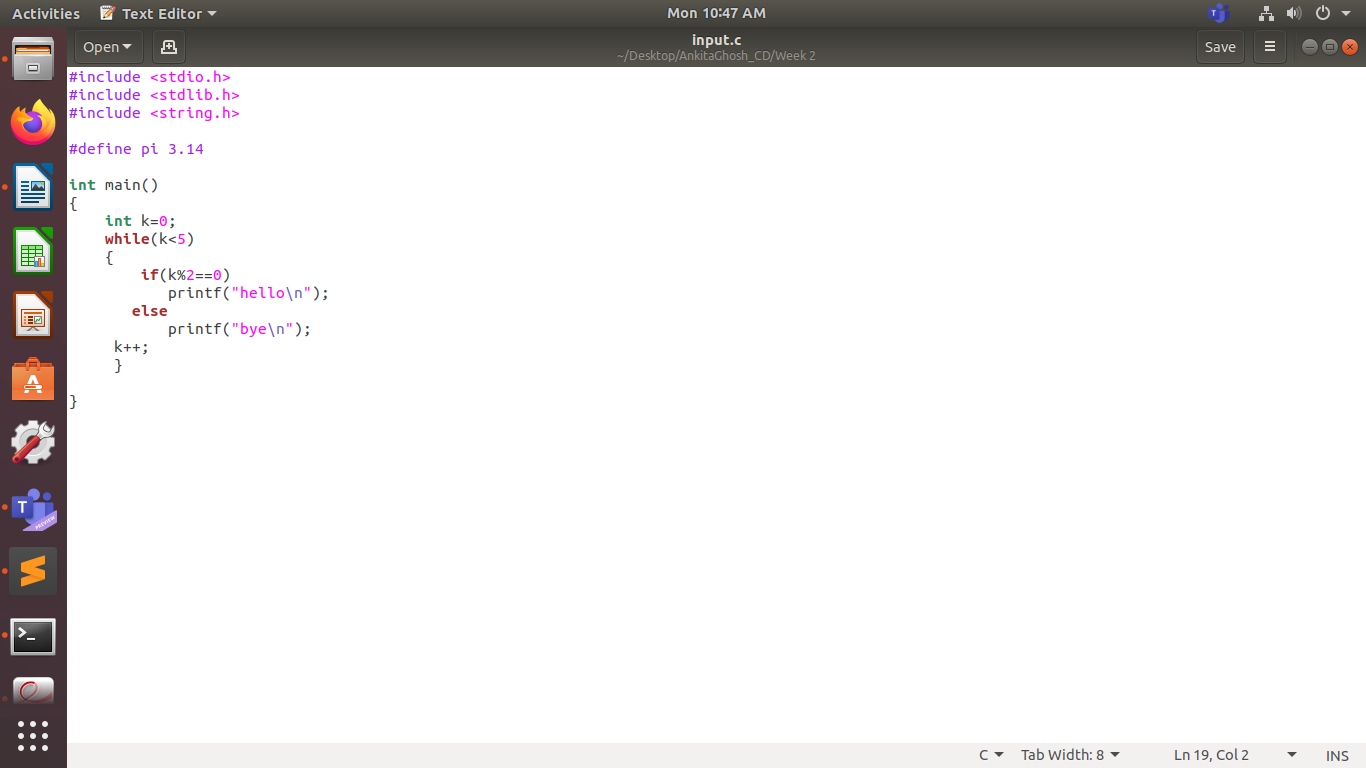
}

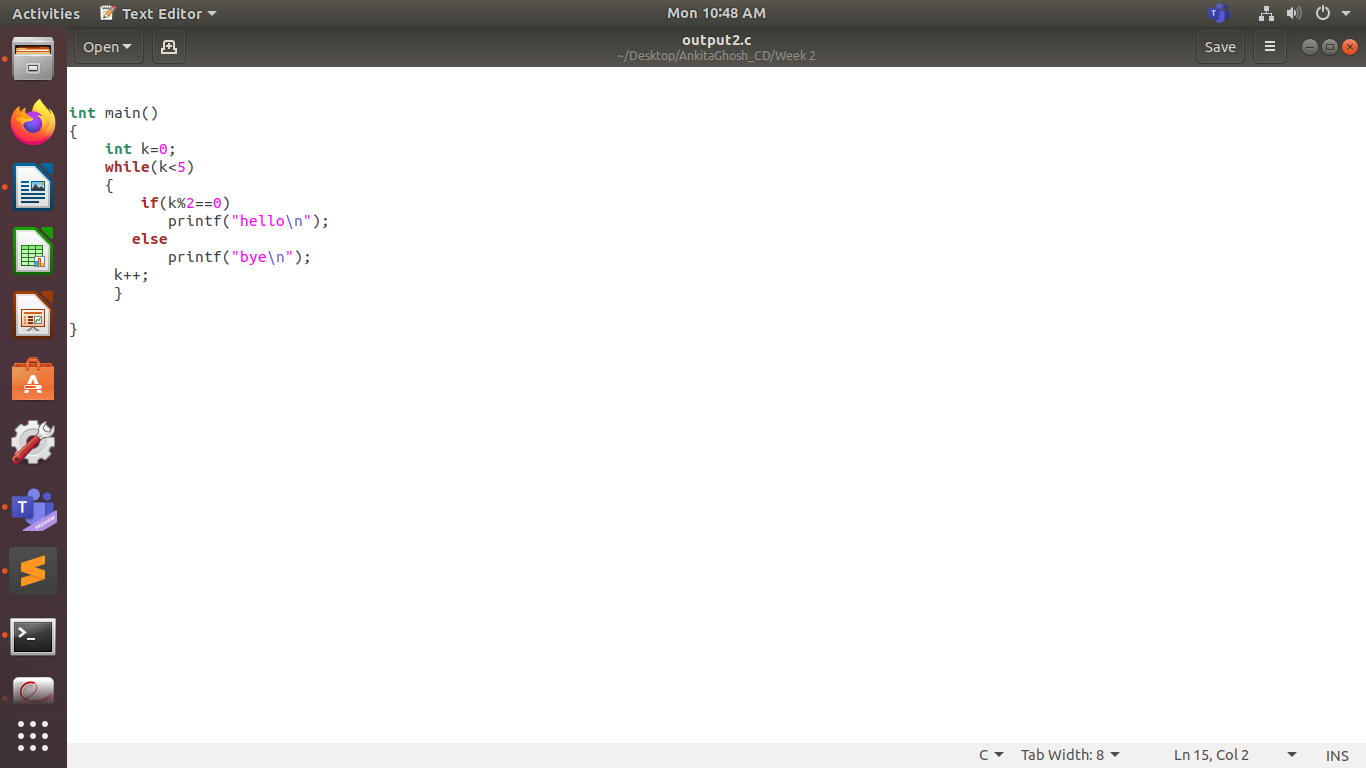
fclose(f1);

fclose(f2);

}

**Output:**





**Q3. That takes C program as input, recognizes all the keywords and prints them in**

**upper case.**

**Program:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

int main()

{

FILE \*fa;

char c;

int n=20; int len=20;

char keyw[20][20] = {"const", "bool", "char", "int","float", "double","return", "for", "while", "do", "switch","if", "else","case", "break","printf"};

fa = fopen("input.c", "r");

if (fa == NULL)

{

printf("Cannot open file \n");

exit(0);

}

c=getc(fa);

printf("Keywords are:\n");

while(c!=EOF)

{

if(isalpha(c))

{

char buf[100];

int x=0;

while(isalpha(c))

{

buf[x++]=c;

c=getc(fa);

}

if(c==' '||c=='\t'||c=='\n'||c=='('||c=='{'||c==';')

{

// printf("entered blank %c\n",c);

buf[x]='\0';

int i;

for(i=0;i<n;i++)

{

if(strcmp(buf,keyw[i])==0)

{

// printf("found match\n");

for(int j=0;j<x;j++)

printf("%c",toupper(buf[j]));

printf("\n");

break;

}

}

}

else

{

// printf("entered not blank %c\n",c);

while(c!=' ' && c!='\t' && c!='\n' && c!='{' && c!=';' && c!=EOF)

c=getc(fa);

}

}

else

{

// printf("entered not blank out %c\n",c);

while(c!=' ' && c!='\t' && c!='\n' && c!=';' && c!='{' && c!=EOF)

c=getc(fa);

}

if(c!=EOF)

c=fgetc(fa);

}

fclose(fa);

printf("Printing Completed\n");

}

**Output:**

