

CD LAB-2 PRELIMINARY SCANNING APPLICATIONS

Name: Ketan Goud

Reg No: 220905260

Section: CSE D- D2

Roll No: 39

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

Code:

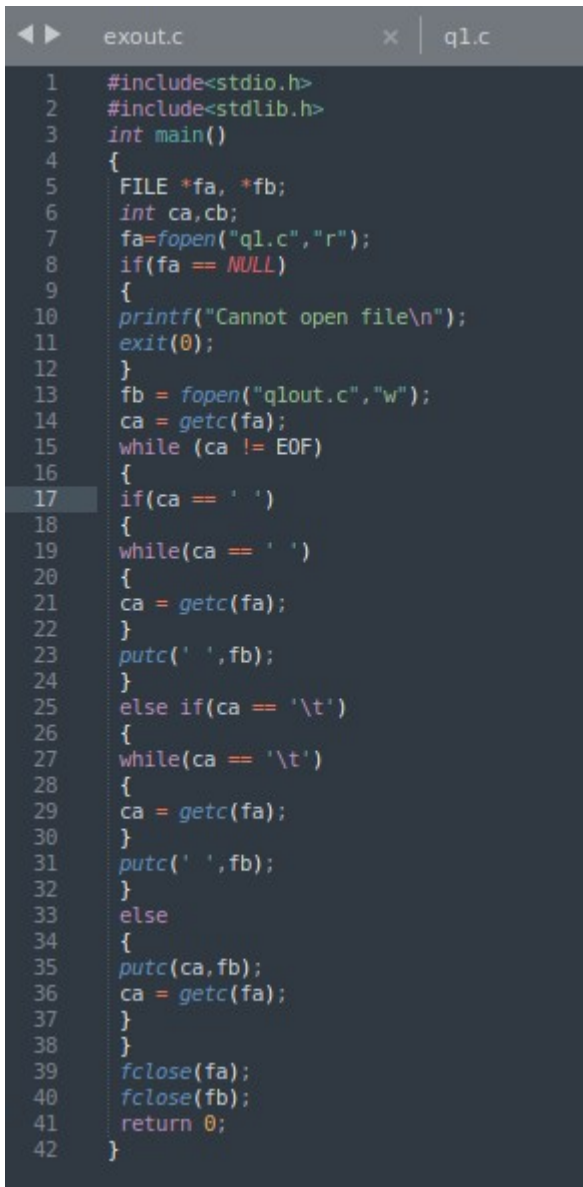
```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    FILE *fa, *fb;
    int ca,cb;
    fa=fopen("q1.c","r");
    if(fa == NULL)
    {
        printf("Cannot open file\n");
        exit(0);
    }
    fb = fopen("q1out.c","w");
    ca = getc(fa);
    while (ca != EOF)
    {
        if(ca == ' ')
        {
            while(ca == ' ')
            {
                ca = getc(fa);
            }
            putc(' ',fb);
        }
        else if(ca == '\t')
        {
            while(ca == '\t')
            {
                ca = getc(fa);
            }
            putc(' ',fb);
        }
        else
        {
            putc(ca,fb);
            ca = getc(fa);
        }
    }
    fclose(fa);
```

```

    fclose(fb);
    return 0;
}

```

Sample Input and Output:



```

1  #include<stdio.h>
2  #include<stdlib.h>
3  int main()
4  {
5      FILE *fa, *fb;
6      int ca,cb;
7      fa=fopen("q1.c","r");
8      if(fa == NULL)
9      {
10         printf("Cannot open file\n");
11         exit(0);
12     }
13     fb = fopen("qlout.c","w");
14     ca = getc(fa);
15     while (ca != EOF)
16     {
17         if(ca == ' ')
18         {
19             while(ca == ' ')
20             {
21                 ca = getc(fa);
22             }
23             putc(' ',fb);
24         }
25         else if(ca == '\t')
26         {
27             while(ca == '\t')
28             {
29                 ca = getc(fa);
30             }
31             putc(' ',fb);
32         }
33         else
34         {
35             putc(ca,fb);
36             ca = getc(fa);
37         }
38     }
39     fclose(fa);
40     fclose(fb);
41     return 0;
42 }

```

2. To discard preprocessor directives from the given input 'C' file.

Code:

```

#include<stdio.h>
#include<stdlib.h>
int main()
{
    FILE *fa, *fb;
    int ca,cb;
    char line[1024];
    fa=fopen("q2.c","r");
    if(fa == NULL)
    {
        printf("Cannot open file\n");
        exit(0);
    }
}

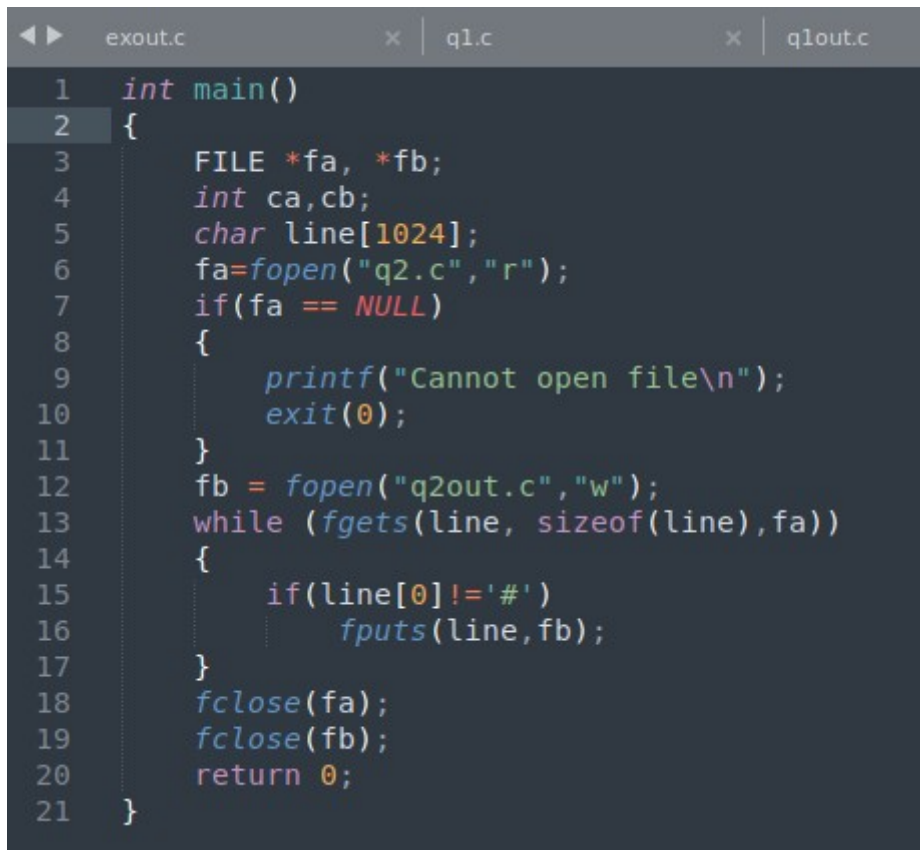
```

```

    }
    fb = fopen("q2out.c","w");
    while (fgets(line, sizeof(line),fa))
    {
        if(line[0]!='#')
            fputs(line,fb);
    }
    fclose(fa);
    fclose(fb);
    return 0;
}

```

Sample Input and Output:



```

1  int main()
2  {
3      FILE *fa, *fb;
4      int ca,cb;
5      char line[1024];
6      fa=fopen("q2.c","r");
7      if(fa == NULL)
8      {
9          printf("Cannot open file\n");
10         exit(0);
11     }
12     fb = fopen("q2out.c","w");
13     while (fgets(line, sizeof(line),fa))
14     {
15         if(line[0]!='#')
16             fputs(line,fb);
17     }
18     fclose(fa);
19     fclose(fb);
20     return 0;
21 }

```

3. That takes C program as input, recognizes all the keywords and prints them in upper case.

Code:

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include <ctype.h>
int check(char word[10])
{
    return (strcmp(word,"else")==0 || strcmp(word, "int")==0 ||
            strcmp(word,"while")==0 || strcmp(word,"if")==0 ||
            strcmp(word,"return")==0 || strcmp(word,"char")==0);
}
void to_uppercase(char word[])
{

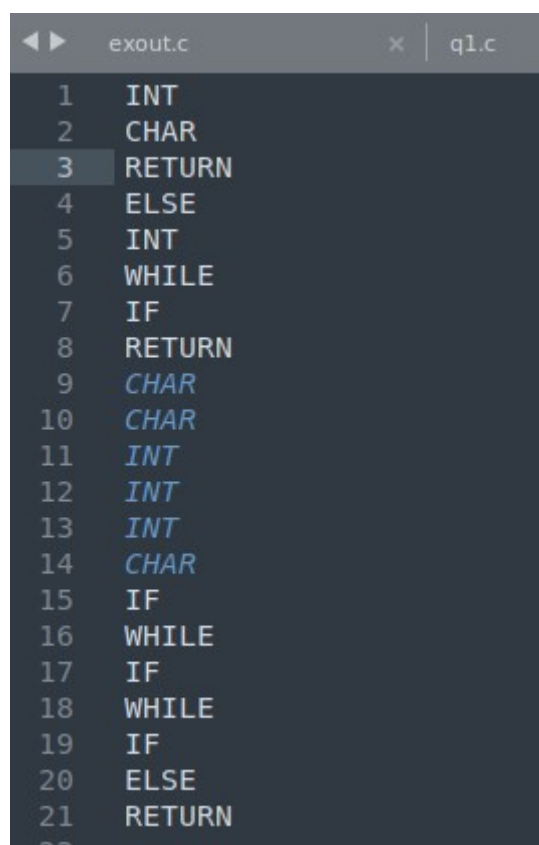
```

```

    for (int i = 0; word[i] != '\0'; i++)
    {
        word[i] = toupper(word[i]);
    }
}
int main()
{
    FILE *fa, *fb;
    int i,ca,cb;
    char word[10];
    fa=fopen("q3.c","r");
    if(fa == NULL)
    {
        printf("Cannot open file\n");
        exit(0);
    }
    fb = fopen("q3out.c","w");
    ca = getc(fa);
    while (ca != EOF)
    {
        if(ca >= 'a' && ca <= 'z')
        {
            i=0;
            while(ca >= 'a' && ca <= 'z')
            {
                word[i]=ca;
                ca = getc(fa);
                i++;
            }
            word[i]='\0';
            if(check(word))
            {
                to_uppercase(word);
                fputs(word,fb);
                putc('\n',fb);
            }
        }
        else
        {
            ca = getc(fa);
        }
    }
    fclose(fa);
    fclose(fb);
    return 0;
}

```

Sample Input and Output:



```
exout.c x | ql.c
1  INT
2  CHAR
3  RETURN
4  ELSE
5  INT
6  WHILE
7  IF
8  RETURN
9  CHAR
10 CHAR
11 INT
12 INT
13 INT
14 CHAR
15 IF
16 WHILE
17 IF
18 WHILE
19 IF
20 ELSE
21 RETURN
22
```