

NAME: AYANIK PAUL

Roll No. 22

D1

LAB 2

Q1) Write a 'C' program

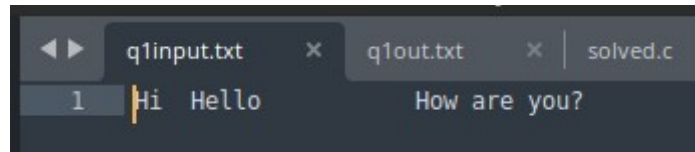
1. That takes a file as input and replaces blank spaces and tabs by single space and writes the output to a file.

CODE:



```
1  #include <stdio.h>
2
3  int main() {
4      FILE *inputFile, *outputFile;
5      char inputFileName[100], outputFileName[100];
6      int currentChar, previousChar = 0;
7      printf("Enter the name of the input file: ");
8      scanf("%s", inputFileName);
9      inputFile = fopen(inputFileName, "r");
10     if (inputFile == NULL) {
11         printf("Cannot open file %s.\n", inputFileName);
12         return 1;
13     }
14     printf("Enter the name of the output file: ");
15     scanf("%s", outputFileName);
16     outputFile = fopen(outputFileName, "w");
17     if (outputFile == NULL) {
18         printf("Cannot open file %s.\n", outputFileName);
19         fclose(inputFile);
20         return 1;
21     }
22     while ((currentChar = getc(inputFile)) != EOF) {
23         if (currentChar == ' ' || currentChar == '\t') {
24             if (previousChar != ' ') {
25                 putc(' ', outputFile);
26                 previousChar = ' ';
27             }
28         } else {
29             putc(currentChar, outputFile);
30             previousChar = currentChar;
31         }
32     }
33
34     // Close the files
35     fclose(inputFile);
36     fclose(outputFile);
37
38     printf("Processing complete. Output written to %s.\n", outputFileName);
39     return 0;
40 }
```

Input file : q1in.txt

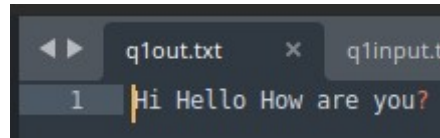


A screenshot of a code editor window. The title bar shows three tabs: 'q1input.txt' (active), 'q1out.txt', and 'solved.c'. The 'q1input.txt' tab contains the text 'Hi Hello' on the first line and 'How are you?' on the second line. A yellow cursor is positioned at the end of the first line.

Terminal

```
student@oslab-02:~/220905128/lab2/q1$ cc q1.c
student@oslab-02:~/220905128/lab2/q1$ ./a.out
Enter the name of the input file: q1input.txt
Enter the name of the output file: q1out.txt
Processing complete. Output written to q1out.txt.
student@oslab-02:~/220905128/lab2/q1$
```

Output file: q1out.txt



A screenshot of a code editor window. The title bar shows two tabs: 'q1out.txt' (active) and 'q1input.txt'. The 'q1out.txt' tab contains the text 'Hi Hello How are you?' on the first line. A yellow cursor is positioned at the end of the first line.

Q2) To discard preprocessor directives from the given input 'C' file.

CODE:

```
q2.c x q2in.c x q2out.c x q3.c
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdbool.h>
4
5  bool is_preprocessor_directive(const char *line) {
6      // Skip leading spaces or tabs
7      while (*line == ' ' || *line == '\t') {
8          line++;
9      }
10     return *line == '#';
11 }
12
13 int main() {
14     char input_file_name[256], output_file_name[256];
15     printf("Enter the input file name: ");
16     scanf("%s", input_file_name);
17     printf("Enter the output file name: ");
18     scanf("%s", output_file_name);
19     FILE *input_file = fopen(input_file_name, "r");
20     FILE *output_file = fopen(output_file_name, "w");
21
22     if (!input_file || !output_file) {
23         printf("Error opening file.\n");
24         return 1;
25     }
26     char line[1024];
27     while (fgets(line, sizeof(line), input_file)) {
28         if (!is_preprocessor_directive(line)) {
29             fputs(line, output_file);
30         }
31     }
32     fclose(input_file);
33     fclose(output_file);
34     printf("Preprocessor directives removed and output written to %s\n", output_file_name);
35     return 0;
36 }
```

Input file: q2in.c

```
q2in.c x q2.c x q2out
1  #include <stdio.h>
2
3  #define gfg 7
4
5  #if gfg > 200
6  #undef gfg
7  #define gfg 200
8  #elif gfg < 50
9  #undef gfg
10 #define gfg 50
11 #else
12 #undef gfg
13 #define gfg 100
14 #endif
15
16 void printValue(int value) { printf("%d", value); }
17
18 int main()
19 {
20     printValue(gfg); // gfg = 50
21     #qq
22     PrintLineNum;
23     printf("#hi");
24     return 0;
25
26 }
```

Terminal:

```
student@oslab-02:~/220905128/lab2/q2$ cc q2.c
student@oslab-02:~/220905128/lab2/q2$ ./a.out
Enter the input file name: q2in.c
Enter the output file name: q2out.c
Preprocessor directives removed and output written to q2out.c
student@oslab-02:~/220905128/lab2/q2$ cd
```

Output file: q2out.c



```
q2out.c  x  q2in.c  x  q2.c
1
2
3
4 void printValue(int value) { printf("%d", value); }
5
6 int main()
7 {
8     printValue(gfg); // gfg = 50
9     PrintLineNum;
10    printf("#hi");
11    return 0;
12
13 }
```

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

CODE:

```
q3.c
1  #include <stdio.h>
2  #include <string.h>
3  #include <ctype.h>
4  #define MAX_KEYWORDS 32
5  const char *keywords[MAX_KEYWORDS] = {
6      "auto", "break", "case", "char", "const", "continue", "default", "do",
7      "double", "else", "enum", "extern", "float", "for", "goto", "if",
8      "int", "long", "register", "return", "short", "signed", "sizeof",
9      "static", "struct", "switch", "typedef", "union", "unsigned", "void",
10     "volatile", "while"
11 };
12 int isKeyword(const char *word) {
13     for (int i = 0; i < MAX_KEYWORDS; i++) {
14         if (strcmp(word, keywords[i]) == 0) {
15             return 1; // It's a keyword
16         }
17     }
18     return 0;
19 }
20 void toUpperCase(char *str) {
21     for (int i = 0; str[i]; i++) {
22         str[i] = toupper(str[i]);
23     }
24 }
25 int main() {
26     FILE *inputFile, *outputFile;
27     char inputFileName[100], outputFileName[100];
28     char word[256];
29     int ch, index = 0;
30     printf("Enter the name of the input C file: ");
31     scanf("%s", inputFileName);
32     inputFile = fopen(inputFileName, "r");
33     if (inputFile == NULL) {
34         printf("Cannot open file %s.\n", inputFileName);
35         return 1;
36     }
37     printf("Enter the name of the output file: ");
38     scanf("%s", outputFileName);
39     outputFile = fopen(outputFileName, "w");
40     if (outputFile == NULL) {
41         printf("Cannot open file %s.\n", outputFileName);
42         fclose(inputFile);
43         return 1;
44     }
45     while ((ch = getc(inputFile)) != EOF) {
46         if (isalnum(ch) || ch == '_' || ch == '-') {
47             word[index++] = ch;
48         } else {
49             if (index > 0) {
```



```

48     } else {
49         if (index > 0) {
50             // End of a word
51             word[index] = '\0';
52             if (isKeyword(word)) {
53                 toUpperCase(word);
54             }
55             fputs(word, outputFile);
56             index = 0;
57         }
58         putc(ch, outputFile);
59     }
60 }
61 if (index > 0) {
62     word[index] = '\0';
63     if (isKeyword(word)) {
64         toUpperCase(word);
65     }
66     fputs(word, outputFile);
67 }
68 fclose(inputFile);
69 fclose(outputFile);
70
71 printf("Processing complete. Keywords have been converted to uppercase in %s.\n", outputFileName);
72 return 0;
73 }

```

Input file: q3in.c

```

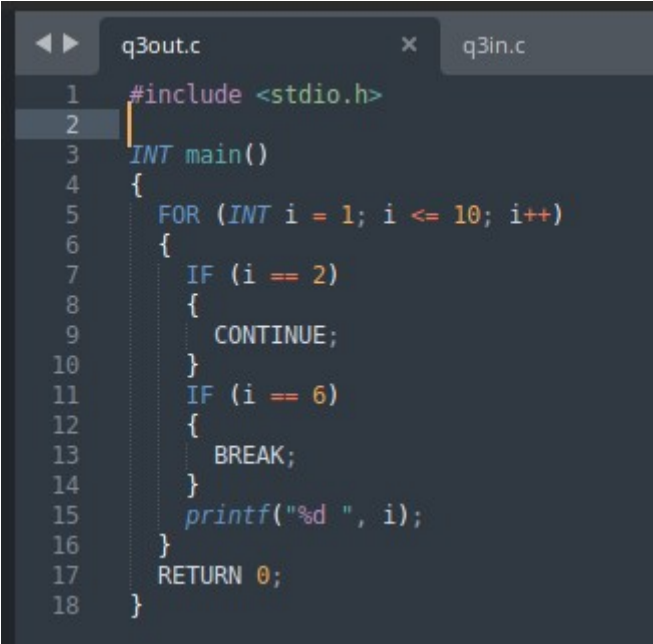
q3in.c  x  q3.c
1  #include <stdio.h>
2
3  int main()
4  {
5      for (int i = 1; i <= 10; i++)
6      {
7          if (i == 2)
8          {
9              continue;
10         }
11         if (i == 6)
12         {
13             break;
14         }
15         printf("%d ", i);
16     }
17     return 0;
18 }

```

Terminal:

```
student@oslab-02:~/220905128/lab2/q3$ cc q3.c
student@oslab-02:~/220905128/lab2/q3$ ./a.out
Enter the name of the input C file: q3in.c
Enter the name of the output file: q3out.c
Processing complete. Keywords have been converted to uppercase in q3out.c.
student@oslab-02:~/220905128/lab2/q3$
```

Output file: q3out.c



```
q3out.c  x  q3in.c
1  #include <stdio.h>
2
3  INT main()
4  {
5      FOR (INT i = 1; i <= 10; i++)
6      {
7          IF (i == 2)
8          {
9              CONTINUE;
10         }
11         IF (i == 6)
12         {
13             BREAK;
14         }
15         printf("%d ", i);
16     }
17     RETURN 0;
18 }
```