

### Question 84049:

Create or find 3 files containing C programs such as the one given below

```
#include <stdio.h>
struct time {
int hour,min;
};
typedef struct {
int flightno;
struct time arrival;
}FlightInfo;

void inp(FlightInfo arr[]){
int i;
for (i=0;i<2;i++){
{
printf("Enter flight no and arrival time:");
scanf("%d %d %d",&arr[i].flightno,&arr[i].arrival.hour,&arr[i].arrival.min);}}
void calculateDelay(FlightInfo arr[], int d){
int i,x;
for (i=0;i<2;i++) {
```

For the three C programs,

- Read the file name from the user and open the file
- Read the C program from the text file and find the
  - number of statements in the program
  - number of for, while and do-while loops in the program

Sample Run:

Program named "flights.cpp" contains:  
53 lines of code  
3 for loops  
2 while loops  
1 do-while loop

Program named "hw.cpp" contains:  
45 lines of code  
1 for loop  
3 while loops  
0 do-while loop

Answer:

hello.c:

```
#include <stdio.h>

int main(void)
{
    for (int i = 0, j = 0; i == 0; i++) {
```

```

        while (j++ == 0) {
            do {
                printf("Hello world\n");
            } while (j == 0)
        }
    }
    return 0;
}

```

stats.c:

```

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

int readline(FILE *fp, char *str, size_t size) {
    int ch = 0;
    char *str1 = str;
    size_t len = 0;
    static const int WIN32_CONSOLE_EOF = 26;

    if (!fp || !str || size == 0)
        return 0;

    while (((len + 1) < size)
        && ((ch = fgetc(fp)) != EOF && ch != WIN32_CONSOLE_EOF))
        *(str1++) = ch;
    *str1 = 0;

    return ch == EOF || ch == WIN32_CONSOLE_EOF;
}

int skipqt(FILE *fp, char qt, int undo) {
    int skip = 0;
    char ch;

    if (undo)
        ungetc(qt, fp);
    if ((ch = fgetc(fp)) == qt)
        while (ch != EOF && !skip) {
            ch = fgetc(fp);
            if (ch == '\\') fgetc(fp), ch = fgetc(fp);
            skip = ch == qt;
        }

    return skip;
}

int skipcom(FILE *fp) {
    int skip = 0;
    char ch;

    while (ch != EOF && !skip)
        skip = ((ch = fgetc(fp)) == '*') && ((ch = fgetc(fp)) == '/');

    return skip;
}

```

```

int matchword(FILE *fp, const char *word, int undo, int skipsp) {
    int match = 1;
    int ch;

    if (skipsp) {
        do { ch = fgetc(fp); }
        while (ch != EOF && (ch < 'a' || ch > 'z') && (ch < 'A' || ch > 'Z'));
        ungetc(ch, fp);
    } else if (undo)
        ungetc(word[0], fp);
    while (match && *word)
        match = *word++ == fgetc(fp);
    match = match && (ch = fgetc(fp)) != EOF && (ch < 'a' || ch > 'z')
        && (ch < 'A' || ch > 'Z') && (ch < '0' || ch > '9');
    if (!match) {
        while (ch != EOF
            && ((ch > 'a' && ch < 'z') || (ch < 'A' && ch > 'Z')))
            ch = fgetc(fp);
    }

    return match;
}

int main(void) {
    char filename[80 + 1];
    FILE *fp;
    int nlines = 0, nfor = 0, nwhile = 0, ndowhile = 0;
    int indowhile = 0, inword = 0;
    int ch = EOF, ch2;

    /* Hint: instead of pressing <Enter>, input an EOF character */
    printf("Program named \"");
    if (getline(stdin, filename, sizeof filename) != 1) {
        printf("\nInput error\n");
        return 1;
    } else if (!(fp = fopen(filename, "r"))) {
        printf("\nFile error\n");
        return 1;
    }
    printf("\n contains:\n");

    /* This is an example of a parser that (only) works with simple programs */
    while (ch2 = ch, (ch = fgetc(fp)) != EOF) nlines += ch == '\n';
    nlines += (ch2 == '\n' || ch2 == EOF);
    clearerr(fp), fseek(fp, 0, SEEK_SET);

    while (ch2 = ch, (ch = fgetc(fp)) != EOF) {
#ifdef TEST
        putchar(ch);
#endif
        if (ch == '\\' || ch == '')
            skipqt(fp, ch, 1), inword = 0;
        else if (ch == '*' && ch2 == '/')
            skipcom(fp), inword = 0;
        else if ((ch == 'f' || ch == 'w' || ch == 'd' || ch == '}')
            && inword == 0) {
            int match = 0;
            if (ch == 'f')

```

```

        match = matchword(fp, "for", 1, 0);
    else if (ch == 'w')
        match = matchword(fp, "while", 1, 0);
    else if (ch == 'd')
        match = matchword(fp, "do", 1, 0);
    else if (ch == '}')
        match = matchword(fp, "while", 0, 1);

    if (match && ch == 'f')
        nfor++;
    else if (match && ch == 'w')
        nwhile++;
    else if (match && ch == 'd')
        indowhile++;
    else if (match && ch == '}' && indowhile)
        indowhile--, ndowhile++;

    inword = 0;
} else {
    if (inword)
        inword = (ch > 'a' && ch < 'z') || (ch < 'A' && ch > 'Z')
            || (ch < '0' && ch > '9');
    else
        inword = (ch > 'a' && ch < 'z') || (ch < 'A' && ch > 'Z');
}
}

printf("%d lines of code\n", nlines);
printf("%d for loop\n", nfor);
printf("%d while loops\n", nwhile);
printf("%d do-while loop\n", ndowhile);

return 0;
}

```

Answer provided by [www.AssignmentExpert.com](http://www.AssignmentExpert.com)