**<NOME DA INSTITUIÇÃO DE ENSINO>**

**<NOME DO INSTITUTO>**

# Teste de Mutação

Nome:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(Em grupo de 2 alunos)**

1. A tabela a seguir apresenta alguns mutantes gerados para o programa count.c. Analise esses mutantes, determinando os mutantes equivalentes e gerando casos de teste para os mutantes não equivalentes. Para os mutantes equivalentes, apresentar uma justificativa.

|  |  |  |
| --- | --- | --- |
| **#** | **Mutação** | Resultado da análise |
| 10 | tlinect=**0** |  |
| 10 | twordct = **-1** |  |
| 12 | i=**0** |  |
| 14 | if (argc > 1 **\*** (... |  |
| 18 | linect = wordct = charct = **1** |  |
| 20 | while ((**inword** = getc(fp)) |  |
| 24 | if (c == ' ' **+** c == '\t' || c == '\n') |  |
| 26 | else if (inword >= 0) |  |
| 26 | else if (inword <= 0) |  |
| 27 | **;** |  |
| 38 | twordct = wordct; |  |
| 40 | } while (++i **<=** argc); |  |
| 42 | printf("%7ld %7ld %7ld total\n", **tlinect**, twordct, tcharct); |  |

.

**Programa count.c**

1 #include <stdio.h>

2

3 main (argc, argv)

4 int argc;

5 char \*argv[];

6 {

7 int c, i, inword;

8 FILE \*fp;

9 long linect, wordct, charct;

10 long tlinect = 1, twordct = 1, tcharct = 1;

11

12 i = 1;

13 do {

14 if (argc > 1 && (fp=fopen(argv[i], "r")) == NULL) {

15 fprintf (stdout, "can't open %s\n", argv[i]);

16 exit (1);

17 }

18 linect = wordct = charct = 0;

19 inword = 1;

20 while ((c = getc(fp)) != EOF) {

21 ++charct;

22 if (c == '\n')

23 ++linect;

24 if (c == ' ' || c == '\t' || c == '\n')

25 inword = 0;

26 else if (inword == 0) {

27 inword = 1;

28 ++wordct;

29 }

30 }

31 printf("%7ld %7ld %7ld", linect, wordct, charct);

32 if (argc > 1)

33 printf(" %s\n", \*argv);

34 else

35 printf("\n");

36 fclose(fp);

37 tlinect += linect;

38 twordct += wordct;

39 tcharct += charct;

40 } while (++i < argc);

41 if (argc > 1)

42 printf("%7ld %7ld %7ld total\n", tlinect, twordct, tcharct);

43 exit(0);

44 }

**Especificação do programa**

**Nome**

count – programa que conta o número de linhas, palavras e caracteres

**Como usar**

count filename [filename...]

**Descrição**

*count* conta o número de linhas, palavras e caracteres de arquivos fornecidos como parâmetros. Palavras são sequências de caracteres separadas por um ou mais espaços, tabulações ou quebras de linha.

Se o arquivo fornecido como argumento não existir, uma mensagem de erro correspondente será impressa e o processamento dos outros arquivos fornecidos como parâmetros continuará. Se nenhum arquivo for fornecido como parâmetro, a contagem será lida a partir da entrada pelo teclado (standard input).

Os valores são calculados para cada arquivo fornecido. A saída impressa mostra os valores da soma de linhas, de palavras, de caracteres e o nome do arquivo. Após mostrar o resultado de todos os arquivos fornecidos, mostra a soma de todos os valores (a palavra “total” é impressa no quarto valor. Se a entrada for lida pelo teclado, o quarto valor (nome do arquivo) não será mostrada.

**Exemplo**

% count datei

84 462 3621 datei