Uso de `fgets()` em programas

Sintaxe: `fgets( variável_destino, qtde_maxima_bytes, stdin )`

Exemplo:

```
char texto[1000];
int c;
...
fgets(texto, 1000, stdin );
```

O conteúdo lido da entrada padrão (determinado por 'stdin' na sintaxe utilizada no exemplo) será armazenado na variável texto. Se o conteúdo da entrada padrão for maior que 1000 caracteres, apenas os primeiros 1000 caracteres serão lidos, no exemplo acima. Considere que tanto caracteres gráficos (como letras, dígitos numéricos e caracteres especiais) como caracteres de controle (como a indicação de um <enter>, por exemplo) são retornados por `fgets`.

Para ler com `fgets()` até o fim de arquivo:

```
while( fgets(texto, 1000, stdin) != NULL )
```

Se você precisar utilizar no mesmo programa `fgets()` e `scanf()`, precisará tomar cuidado, pois o `scanf()` deixa resíduos no buffer de entrada que o `fgets()` acaba lendo. Pra evitar esse problema, após cada `scanf()` faça a limpeza desse lixo, por meio da rotina `LimpaBuffer()` mostrada a seguir, que você deverá criar em seu programa:

```
void LimpaBuffer()
/* Limpa buffer de entrada para fgets */
{     while(getchar()!='\n');
      return;
}
...
scanf("%d", &N);
LimpaBuffer();
...
fgets(texto, 1000, stdin);
...
```

Como a *string* lida com `fgets()` incorpora caracteres de quebra de linha presentes no texto, você poderá querer descarta-los, pois geralmente não são um conteúdo útil para o seu programa. Nesse caso, faça:

```
fgets(texto , 1000, stdin);
for( c=0; texto[c] >= ' ' || texto[c] == '\t'; c++ );
texto[c]= '\0';
```