

1. A entrada do programa consiste em uma palavra com até 20 caracteres composta somente por . e x, onde o primeiro representa um espaço vazio e o segundo representa uma bomba. Depois o programa deve ler um inteiro representando um índice na palavra (começando de 0) e escrever na tela **bum!** caso nesse índice haja uma bomba ou escrever na tela um inteiro representando quantas bombas há na adjacência do índice em questão.

Input	Output
..xx..x..x.. 0	0
..xx..x..x.. 1	1
..xx..x..x.. 2	bum!

2. Crie um programa que leia 2 palavras (até 30 caracteres cada) e que escreva na tela uma string alternando as letras de ambas as strings. Caso não possuam o mesmo tamanho, as letras restantes da string maior devem ser escritas ao final da string resultante. Por exemplo, ao misturar "teste" e "algo", obtém-se "taelsgtoe". Já ao misturar "asa" e "inconstitucionalidade" obtém-se "aisnaconstitucionalidade".

Exemplos:

Input	Output
teste algo	taelsgtoe
asa inconstitucionalidade	aisnaconstitucionalidade
rsldo eutd	resultado

3. > Escreva um programa em C que leia uma frase de até 200 caracteres, altere todos os caracteres de acordo com essa escrita *leet*: a → 4, e → 3, i → 1, s → 5, t → 7 e, em seguida, escreva-a na tela.

Exemplo:

Input	Output
Esta eh uma string	E574 3h um4 57r1ng

4. Escreva um programa em C que leia uma frase de até 200 caracteres e escreva na tela a quantidade de palavras da string. Define-se aqui palavra como qualquer sequência de caracteres diferentes de espaços.

Exemplos:

Input	Output
Quantas palavras ha nessa frase?	5
Esta-eh-uma str:ng	2
Este eh um ponto .	5

5. > Escreva um programa em C que leia uma palavra de até 20 caracteres e em seguida escreva se essa palavra representa uma placa válida (S ou N). Uma palavra é uma placa válida se contiver somente e na sequência: 3 letras maiúsculas, hífen e 4 dígitos numéricos.

Exemplo:

Input	Output
AGH-3201	S
AGh-3201	N
AG3201	N

6. Escreva um programa que verifica se uma string contém um código válido. Para ser tal, a string deve estar exatamente no formato: AB-X. Ou seja 4 caracteres, sendo nessa ordem: 2 dígitos, traço e 1 dígito verificador. Um dígito verificador têm como objetivo evitar fraudes ou erros de digitação/transmissão. O dígito X deve ser o resto da divisão de $2 * A + 3 * B$ por 7 (lembre-se de que é o valor inteiro do dígito e não o valor pela tabela ASCII). O programa deve ler uma palavra de até 20 caracteres e escrever na tela S caso seja um código válido ou N caso contrário.

Exemplos:

Input	Output
35-0	S
35-00	N
74-4	N

7. Atoi é uma função em C que transforma uma string em número inteiro. Escreva a seguinte função obterInteiro que recebe como parâmetro um vetor **s** de caracteres (ou seja, uma string) e que retorna o inteiro contido da string observando essas regras:

- pode haver caracteres diferentes de dígitos numéricos, mas eles são ignorados
- o número pode ser negativo, desde que o - seja o **primeiro** caractere
- caso não haja dígito numérico, o resultado deve ser 0

```
int obterInteiro(char s[])
```

A função main deve ler do usuário uma palavra de até 30 caracteres e escrever na tela o inteiro contido na string fazendo uso da função obterInteiro. Por exemplo, se a string for "tes100te", a função deve retornar 100 e se a string for "a-s1tri2g" deve retornar 12.

Exemplos:

Input	Output
3491	3491
0001k-jt3	13
-gm12	-12

8. > Escreva a seguinte função que recebe como parâmetro um vetor **s** de caracteres (ou seja, uma string) e que retorna a quantidade de dígitos numéricos (0 a 9) nessa string:

```
int qtdDigitosNumericos(char s[])
```

A função main deve ler do usuário uma palavra de até 20 caracteres e escrever na tela a quantidade de dígitos numéricos fazendo uso da função qtdDigitosNumericos.

Exemplos:

Input	Output
tes100te	3
as1tri2g	2

9. Crie uma função que **modifica** uma string substituindo todos os caracteres **x** pelo caractere **y**:

```
void substituir(char s[], char x, char y);
```

Por exemplo, ao chamar substituir(palavra, 'a', 'e'), sendo palavra **lata**, torna-se **lete**.

A função *main* deve ler do usuário uma palavra de até 20 caracteres, ler o caractere que será substituído, ler o caractere que o substituirá e, em seguida, escrever na tela a string resultante. A string é separada dos caracteres por um único espaço. Utilize o seguinte código para fazer a leitura:

```
char s[21];
char x, y;
scanf("%s", s);
getchar();
scanf("%c %c", &x, &y);
```

Input	Output
Teste t v	Tesve

Nota: você não poderá chamar a função passando uma string literal como primeiro argumento (exemplo: substituir("lata", 'a', 'e')). Motivos: primeiro "lata" está na memória, mas não pode ser modificada (ocasiona falha de segmentação se tentar) e, segundo, mesmo que fosse possível alterar, teríamos que vinculá-la a alguma variável para escrevê-la posteriormente. O ideal portanto é na função main declarar uma string (como no código sugerido).