OBI 2008 - Nível 2: Fase 1

Telefone

Prof. Edson Alves

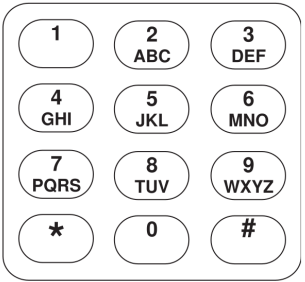
Faculdade UnB Gama

As primeiras redes públicas de telefonia foram construídas pela AT&T; no começo do século XX. Elas permitiam que seus assinantes conversassem com a ajuda de uma telefonista, que conectava as linhas dos assinantes com um cabo especial.

Essas redes evoluíram muito desde então, com a ajuda de vários avanços tecnológicos. Hoje em dia, essas redes atendem centenas de milhões de assinantes; ao invés de falar diretamente com uma telefonista, você pode simplesmente discar o número da pessoa desejada no telefone.

Cada assinante recebe um número de telefone – por exemplo, 55-98-234-5678. Qualquer pessoa que discar esse número consegue então falar com a pessoa do outro lado da linha. Os hifens no número de telefone são só para facilitar a leitura, e não são discados no telefone.

Para que fique mais fácil de se lembrar de um número de telefone, muitas companhias divulgam números que contém letras no lugar de dígitos. Para convertê-los de volta para dígitos, a maioria dos telefones tem letras nas suas teclas:



Ao invés de discar uma letra, disca-se a tecla que contém aquela letra. Por exemplo, se você quiser discar o número 0800-FALE-SBC, você na realidade discaria 0800-3253-722.

A sua avó tem reclamado de problemas de vista – em particular, ela não consegue mais enxergar as letrinhas nas teclas do telefone, e por isso queria que você fizesse um programa que convertesse as letras em um número de telefone para dígitos.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A entrada é composta de apenas uma linha, contendo o número de telefone que deve ser traduzido. O número de telefone contém entre 1 e 15 caracteres, que podem ser dígitos e '0' a '9', letras de 'A' a 'Y' e hífens ('–').

Saída

Seu programa deve imprimir, na saída padrão, uma única linha, contendo o número de telefone com as letras convertidas para dígitos. Hífens no número telefone devem ser mantidos no número de telefone de saída.

Exemplo de entrada e saída

Exemplo de entrada e saída

```
M1S-TU-R4
```

Exemplo de entrada e saída

M1S-TU-R4

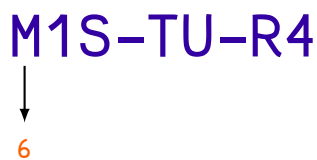
Telefone com letras, números e hífens

Exemplo de entrada e saída

```
M1S-TU-R4
```

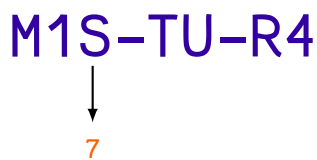
Exemplo de entrada e saída

M1S-TU-R4



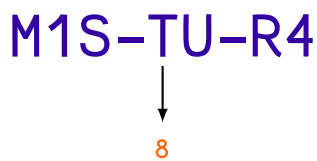
Exemplo de entrada e saída

M1S-TU-R4



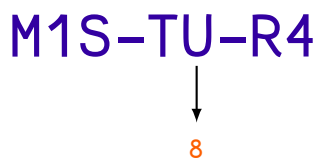
Exemplo de entrada e saída

M1S-TU-R4



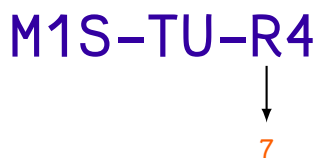
Exemplo de entrada e saída

M1S-TU-R4



Exemplo de entrada e saída

M1S-TU-R4



Solução

Solução

⋆ Como o número de caracteres N não é dado na entrada, o telefone deve ser lido como uma string

Solução

⋆ Como o número de caracteres N não é dado na entrada, o telefone deve ser lido como uma string

⋆ A função `strlen()` pode ser usada para computar o valor de N

Solução

⋆ Como o número de caracteres N não é dado na entrada, o telefone deve ser lido como uma string

⋆ A função `strlen()` pode ser usada para computar o valor de N

⋆ O problema consiste em substituir os caracteres alfabéticos pelos equivalentes numéricos

Solução

⋆ Como o número de caracteres N não é dado na entrada, o telefone deve ser lido como uma string

⋆ A função `strlen()` pode ser usada para computar o valor de N

⋆ O problema consiste em substituir os caracteres alfabéticos pelos equivalentes numéricos

⋆ Os demais caracteres devem permanecer inalterados

Solução

⋆ Como o número de caracteres N não é dado na entrada, o telefone deve ser lido como uma string

⋆ A função `strlen()` pode ser usada para computar o valor de N

⋆ O problema consiste em substituir os caracteres alfabéticos pelos equivalentes numéricos

⋆ Os demais caracteres devem permanecer inalterados

⋆ Os equivalentes numéricos podem ser determinados por uma série de **`if`**/**`else`** em cascata

```c
#include <stdio.h>


int main()
{
```

```c
    return 0;
}
```

```c
#include <stdio.h>


int main()
{
    char s[16];
    scanf("%s", s);
    return 0;
}
```

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

int main()
{
    char s[16];
    scanf("%s", s);

    int N = strlen(s);

    return 0;
}
```

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

int main()
{
    char s[16];
    scanf("%s", s);

    int N = strlen(s);

    for (int i = 0; i < N; ++i)
    {
        }

        return 0;
    }
```

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

int main()
{
    char s[16];
    scanf("%s", s);

    int N = strlen(s);

    for (int i = 0; i < N; ++i)
    {
        if ('A' <= s[i] && s[i] <= 'C')
            s[i] = '2';
    }

    return 0;
}
```

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

int main()
{
    char s[16];
    scanf("%s", s);

    int N = strlen(s);

    for (int i = 0; i < N; ++i)
    {
        if ('A' <= s[i] && s[i] <= 'C')
            s[i] = '2';
        else if ('D' <= s[i] && s[i] <= 'F')
            s[i] = '3';
    }

    return 0;
}
```

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

int main()
{
    char s[16];
    scanf("%s", s);

    int N = strlen(s);

    for (int i = 0; i < N; ++i)
    {
        if ('A' <= s[i] && s[i] <= 'C')
            s[i] = '2';
        else if ('D' <= s[i] && s[i] <= 'F')
            s[i] = '3';
        else if ('G' <= s[i] && s[i] <= 'I')
            s[i] = '4';
        else if ('J' <= s[i] && s[i] <= 'L')
            s[i] = '5';
        else if ('M' <= s[i] && s[i] <= 'O')
            s[i] = '6';
        else if ('P' <= s[i] && s[i] <= 'S')
            s[i] = '7';
        else if ('T' <= s[i] && s[i] <= 'V')
            s[i] = '8';
        else if ('W' <= s[i] && s[i] <= 'Z')
            s[i] = '9';
    }

    return 0;
}
```

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

int main()
{
    char s[16];
    scanf("%s", s);

    int N = strlen(s);

    for (int i = 0; i < N; ++i)
    {
        if ('A' <= s[i] && s[i] <= 'C')
            s[i] = '2';
        else if ('D' <= s[i] && s[i] <= 'F')
            s[i] = '3';
        else if ('G' <= s[i] && s[i] <= 'I')
            s[i] = '4';
        else if ('J' <= s[i] && s[i] <= 'L')
            s[i] = '5';
        else if ('M' <= s[i] && s[i] <= 'O')
            s[i] = '6';
        else if ('P' <= s[i] && s[i] <= 'S')
            s[i] = '7';
        else if ('T' <= s[i] && s[i] <= 'V')
            s[i] = '8';
        else if ('W' <= s[i] && s[i] <= 'Z')
            s[i] = '9';
    }

    printf("%s\n", s);

    return 0;
}
```

Bônus

Bônus

- Há uma solução alternativa para este problema

Bônus

- Há uma solução alternativa para este problema

- Ela é baseada no código que um caractere ocupa na tabela ASCII e um vetor auxiliar

Bônus

- Há uma solução alternativa para este problema
- Ela é baseada no código que um caractere ocupa na tabela ASCII e um vetor auxiliar
- As letras estão em ordem crescente na tabela ASCII

Bônus

- Há uma solução alternativa para este problema
- Ela é baseada no código que um caractere ocupa na tabela ASCII e um vetor auxiliar
- As letras estão em ordem crescente na tabela ASCII
- Assim a posição da letra c no alfabeto pode ser obtida pela diferença `(c - 'A')`

Bônus

⋆ Há uma solução alternativa para este problema

⋆ Ela é baseada no código que um caractere ocupa na tabela ASCII e um vetor auxiliar

⋆ As letras estão em ordem crescente na tabela ASCII

⋆ Assim a posição da letra `c` no alfabeto pode ser obtida pela diferença `(c - 'A')`

⋆ O vetor auxiliar `keyboard` guarda, na i-ésima posição, o número correspondente à i-ésima letra maiúscula no teclado do telefone.

```c
#include <stdio.h>

#include <string.h>

int main()
{
    char s[16];
    scanf("%s", s);

    int N = strlen(s);









    return 0;
}
```

```c
#include <stdio.h>

#include <string.h>

int main()
{
    char s[16];
    scanf("%s", s);

    int N = strlen(s);

    char keyboard [27] = "22233344455566677778889999";









    return 0;
}
```

```c
#include <stdio.h>

#include <string.h>

int main()
{
    char s[16];
    scanf("%s", s);

    int N = strlen(s);

    char keyboard [27] = "22233344455566677778889999";

    for (int i = 0; i < N; ++i)




    return 0;
}
```

```c
#include <stdio.h>
#include <ctype.h>
#include <string.h>

int main()
{
    char s[16];
    scanf("%s", s);

    int N = strlen(s);

    char keyboard [27] = "22233344455566677778889999";

    for (int i = 0; i < N; ++i)
        if (isalpha(s[i]))
            s[i] = keyboard[s[i] - 'A'];



    return 0;
}
```

```c
#include <stdio.h>
#include <ctype.h>
#include <string.h>

int main()
{
    char s[16];
    scanf("%s", s);

    int N = strlen(s);

    char keyboard [27] = "22233344455566677778889999";

    for (int i = 0; i < N; ++i)
        if (isalpha(s[i]))
            s[i] = keyboard[s[i] - 'A'];

    printf("%s\n", s);

    return 0;
}
```