Lista de Exercícios

Leitura de Código

Questão 01 [2xp] - teste o código abaixo para a entrada 5 e 8. O que o programa faz?

```c
#include <stdio.h>

int main(){
    int num, casos;
    int i;
    unsigned long long fib[60];

    fib[0] = 0;
    fib[1] = 1;

    for(i=2; i<=60; i++){
        fib[i] = fib[i-1]+fib[i-2];
    }

    scanf("%d", &casos);

    for(i=0; i<casos; i++){
        scanf("%d", &num);

        printf("Fib(%d) = %llu\n", num, fib[num]);
    }

    return 0;
}
```

Questao 02 [2xp] - teste o código abaixo para as strings Skywalker, walker e droids. O que o programa faz?

```c
int shoryuken(char* st, char* st2){
       int t1 = tamanhoString(st);
       int t2 = tamanhoString(st2);
       int i, j;
       int ok = 0;


       if(t2 > t1) return 0;


       for(i=0; i<=t1-t2; i++){
              for(j=0; j<t2; j++){
                     if(st[i+j] != st2[j]){
                            ok = 1;
                            break;
                     }
```

```
            }

            if(ok == 0)
                  return 1;
            else
                  ok = 0;
      }
      return 0;
}
```

Ache o Erro

Questão 01 [3xp] - A função abaixo deveria concatenar duas strings, mas não funciona muito bem. Qual(is) o(s) problema(s)?

```
void juntaString(char* st1, char* st2, char* cat){
      int t1 = strlen(st1);
      int t2 = strlen(st2);
      int i, j;
      for(i=0; i<t1; i++)
            cat[i] = st1[i];
      for(i=0; i<t2; i++)
            cat[i] = st2[i];
}
```

Complete o Código

Questão 01 [2xp] - A função abaixo compara duas strings em função da ordem lexográfica delas. Complete os comandos de forma que ela retorne -1 se a string st1 for menor que a st2, 0 se forem iguais e 1 se st1 for maior que st2. Lembrando que menor = aparece primeiro na ordem lexográfica e maior = aparece depois na ordem lexográfica.

```
int comparaString(char* st1, char* st2){
      int i;
      int t1 = strlen(st1);
      int t2 = strlen(st2);

      if(t1 == t2){
            comandos
      else if(t1 > t2){
            comandos
      }
```

```
        else{
                comandos
        }
}
```

Problemas para Resolver

Questão 01 [5xp]

Charles está com uma pilha de projetos para corrigir e não tem mais tempo antes de fechar a disciplina! Como os relatórios foram bastante extensos (graças a grande capacidade dissertativa dos alunos), o tamanho médio de um trabalho estava sendo de 125 páginas! Por sorte, Charles pediu para que os alunos iniciassem o trabalho fazendo um resumo, e decidiu que faria a sua correção através dele.

Mesmo assim, ler todos os resumos demoraria muito tempo, e Charles tem um campeonato de Starcraft 2 para participar. Dessa forma, ele resolveu adotar uma heurística para facilitar a sua vida. Caso o resumo apresente uma determinada quantidade de palavras que comecem e acabem com uma determinada letra, e um número mínimo de palavras, a nota seria 10! Caso o resumo não tenha o número mínimo de palavras, a nota seria 2. Caso tivesse o número mínimo de palavras mas não atingisse os limites de palavras que começam e terminam com letras específicas, nota 5. Caso o número de palavras e um dos critérios seja atendido, nota 7,5.

Por exemplo: imagine que a redação tivesse um número mínimo de 10 palavras, sendo que devem aparecer 3 que começam com a letra A e 2 que terminam com a letra R. Então teríamos as seguintes notas para as redações abaixo:

Anakyn Skywalker era acima de todos os outros jedis da ordem em habilidade, embora a fúria ardesse como um calor maligno.

Nota 10: mais de 10 palavras, 3 ou mais palavras começando com A (Anakyn, acima, a, ardesse), 2 ou mais palavras terminando em R (Skywalker, calor)

É nós brother!

Nota 2: menos de 10 palavras

América e ABC sempre disputam para ser o melhor do estado!

Nota 5: mais de 10 palavras, 2 palavras começando com A, 1 terminando com R
América e ABC sempre disputam para ser o melhor do estado forever!

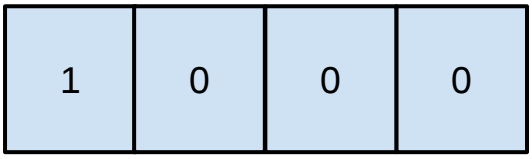
Nota 7.5: mais de 10 palavras, 2 começando com A e 2 terminando com R.

Escreva um programa que, dada uma redação, escreva uma linha de Nota como a contida nos exemplos anteriores. A redação será composta por um texto em múltiplas linhas, onde nenhuma linha terá mais do que 500 letras.

Questão 02 [6xp]

Marcel está com um problema no mínimo inusitado na coordenação dos cursos técnicos do IMD: o número de alunos aumentou exponencialmente! Atualmente, o curso conta com 120387562873465827345230478562093846058273039845019346507263405723049853456345634563456345645 alunos, e está difícil acompanhar todos eles. Como o valor é muito grande para usar calculadoras convencionais, Marcel pediu que os alunos de ITP fizessem uma calculadora para resolver o seu problema!

Para isso, os alunos de ITP vão fazer uma calculadora usando a ideia de BIG NUMBER, ou representação de números utilizando um vetor. Apesar do nome bonito, a ideia é muito simples: cada posição do vetor representa um dígito do número! Então para o número 1000, temos o seguinte vetor:



Faça um programa que implemente as quatro operações básicas (soma, subtração, multiplicação e divisão), para que Marcel possa manter controle dos alunos do técnico! Para implementação, é garantido que o número de alunos nunca ultrapassará a ordem de 10^{10000} alunos.

Questão 03 [2xp] - URI 1332

Seu irmão mais novo aprendeu a escrever apenas um, dois e três, em Inglês. Ele escreveu muitas dessas palavras em um papel e a sua tarefa é reconhecê-las. Nota-se que o seu irmão mais novo é apenas uma criança, então ele pode fazer pequenos erros: para cada palavra, pode haver, no máximo, uma letra errada. O comprimento de palavra é sempre correto. É garantido que cada palavra que ele escreveu é em letras minúsculas, e cada palavra que ele escreveu tem uma interpretação única.

A primeira linha contém o número de palavras que o seu irmão mais novo escreveu. Cada uma das linhas seguintes contém uma única palavra com todas as letras em minúsculo. As palavras satisfazem as restrições acima: no máximo uma letra poderia estar errada, mas o comprimento da palavra está sempre correto. Haverá, no máximo, 1000 palavras de entrada.

Para cada caso de teste, imprima o valor numérico da palavra.

Entrada	**Saída**
3 owe	1

too	2
theee	3

Questão 04 [3xp] - URI 1607

É dado na entrada uma string A e outra B. Em uma operação você pode escolher uma letra da primeira string e avançar esta letra. Avançar uma letra significa transformá-la na próxima letra do alfabeto, veja que a próxima letra depois de z vem a letra a novamente!

Por exemplo, podemos transformar a string ab em bd em no mínimo 3 operações: ab -> bb -> bc -> bd. Podemos aplicar operações nas letras em qualquer ordem, outra possibilidade seria: ab -> ac -> bc -> bd.

Dadas as duas strings, calcule o mínimo número de operações necessárias para transformar a primeira na segunda.

Na primeira linha terá um inteiro **T** (**T** = 100) indicando o número de casos de teste.

Para cada caso, na única linha teremos as duas strings **A** ($1 \leq |\mathbf{A}| \leq 100$* ou $1 \leq |\mathbf{A}| \leq 10^4$** - sendo que |**A**| significa o tamanho da string **A**) e **B** (|**B**| = |**A**|* ou |**B**| = |**A**|**) separadas por um espaço. Ambas as strings são compostas por letras do alfabeto minúsculas apenas e são do mesmo tamanho.

Para cada caso imprima o número mínimo de operações.

Entrada	**Saída**
3	
ab bd	3
abc abc	0
abcdefghiz aaaaaaaaaa	173

Questão 05 [2xp] - URI 1168

João quer montar um painel de leds contendo diversos números. Ele não possui muitos leds, e não tem certeza se conseguirá montar o número desejado. Considerando a configuração dos leds dos números abaixo, faça um algoritmo que ajude João a descobrir a quantidade de leds necessário para montar o valor.

A entrada contém um inteiro **N**, **$(1 \leq N \leq 1000)$** correspondente ao número de casos de teste, seguido de **N** linhas, cada linha contendo um número **$(1 \leq V \leq 10^{100})$** correspondente ao valor que João quer montar com os leds.

Para cada caso de teste, imprima uma linha contendo o número de leds que João precisa para montar o valor desejado, seguido da palavra "leds".

Entrada	**Saída**
3	
115380	27 leds
2819311	29 leds
23456	25 leds