2004/05: Programação Imperativa / Programação de Computadores
Época normal, parte II; tempo da prova (2 partes): 1h:30 m

Nome:	
NMEC:	**BInº:**

Notas. (i) Não pode usar funções relacionadas com "strings" como `strlen`, `strcpy`, etc. (ii) A simplicidade e a eficiência da sua resposta são importantes! (iii) Como é evidente, pode haver interesse em utilizar uma ou mais funções auxiliares.

1. Escreva uma função que imprime uma estatística das classificações de um exame. Os valores que as classificações podem ter são inteiros compreendidos entre 0 e 20. A função

   `void tabela(int n, int v[])`

   recebe como dados as classificações `v` de `n` alunos (essas classificações são `v[0]`,`v[1]`,..., `v[n-1]`). A função deve imprimir o número de alunos com cada classificação possível desde que esse número não seja nulo. Esta informação deve ser apresentada por ordem decrescente de classificações *com o formato indicado no exemplo.*

   Exemplo. Dados: `n`=10, `v[]` = `{8,16,4,18,10,10,8,8,15,16}`
   Resultado esperado:

```
18   valores:   1  aluno
16   valores:   2  alunos
15   valores:   1  aluno
10   valores:   2  alunos
 8   valores:   3  alunos
 4   valores:   1  aluno
```

Resposta

```
#include <stdio.h>
#define NVAL 21

// Nota: só era pedida esta função!
void tabela(int n, int v[]){
  int i,c[NVAL]={0};
  for(i=0;i<n;i++)
    c[v[i]]++;
  for(i=NVAL-1;i>=0;i--)
    if(c[i])
      printf("%3d %10s: %5d %s\n",
              i,i==1?"valor":"valores",c[i],c[i]==1?"aluno":"alunos");
}


int main(){
  int n=10, v[] = {1,16,4,18,10,10,8,8,15,16};
  tabela(n,v);
  return 0;
}
```

Continua no verso

2. Escreva uma função

```
int compr(char s[], char t[])
```

que retorna o comprimento da maior sub-palavra comum aos "strings" `s` e `t`.
Exemplos.

s	t	sub-palavras comuns máximas	valor retornado
cxlalala	claro	la	2
lalala	escuro	-	0
aaab	aabbb	aab	3
abcd	xabybc	ab, bc	2

Notas. Não pode utilizar outros vectores (ou "strings"). Não pode usar funções relacionadas com "strings" como `strlen`, `strpos`, etc. Como é óbvio, a função que escrever não deve incluir qualquer instrução de entrada ou saída de dados.

RESPOSTA
A solução apresentada é muito simples. Note que apenas se pedia a função `compr` (linhas `1` a `14`); incluimos o `main` para que o aluno possa testar o programa.

```
1   //-- para todos os i, j: determina-se o comprimento
2   //   da maior subsequência comum s[i...] e t[j...]
3   int compr(char s[], char t[]){
4     int i,j,max=0;
5     for(i=0;s[i];i++)
6       for(j=0;t[j];j++){
7         int a=i,b=j,c=0;
8         while(s[a] && t[b] && s[a]==t[b]){
9           a++;b++;c++;
10        }
11        if(c>max) max=c;
12      }
13    return max;
14  }

15  #define MAX 100
16  int main(){
17    char s[MAX],t[MAX];
18    while(1){
19      printf("s? "); scanf("%s",s);
20      printf("t? "); scanf("%s",t);
21      printf("Max comum: %d\n",compr(s,t));
22       return 0;
23    }
24  }
```

```
Nota: Alguns erros comuns nas respostas a esta questão:
  - Não reinicializar o contador do comprimento das palavras comuns
    (só é iniciado em 0 uma vez), c no exemplo.
  - Avançar demasiado os índices dos 2 "strings" o que faz a função
    retornar resultados errados por exemplo com
      s = "aaab"
      t = "aaaaaaaaab"
    Neste caso a sub-palavra máxima é encontrada com os índices 0
    (de s) e 6 (de t).
  - Não testar o fim simultâneo dos "strings" usando por exemplo um
    ciclo como
      while(s[m]==t[n]) {cont++; m++, n++,}
    O que acontece se s[m]==t[n]=='\0'?
```