

- Cotação de cada grupo de questões: 3, 4.5, 5.5, 5.5
- Responda às questões de forma clara e concisa. Nas respostas pode usar qualquer função definida no enunciado mesmo que não a tenha implementado.

1. Em cada uma das alíneas, diga o que escreve cada fragmento de código dado.

- a) 

```
int i= 10, j=8, k=4;
if ((i+j)%k==0 || (i*j)%k==0)
    printf("%d %d\n",i+j,i*j);
else
    printf("%d\n",k);
```
- b) 

```
int main() {
    int i, n= 3;
    int v[][3] = {{1,2,3},
                  {1,2,3},
                  {1,2,3}};
    for (i= 0; i < n; i++)
        printf("%d %d\n",v[i][i],v[i][n-1-i]);
    return 0;
}
```
- c) 

```
int main() {
    int a= 25, b= 5, c;
    c= func(a,b);
    printf("(%d,%d) --> %d",a,b,c)
    return 0;
}

int func(int a, int b) {
    while(a!=b) {
        if (a>b)
            a= a-b;
        else
            b= b-a;
    }
    return a;
}
```

2. Suponha que num programa é necessário efectuar a leitura e análise de uma sequência de caracteres lida do *standard input* e terminada pelo caracter mudança de linha. A sequência deveria ser constituída por duas partes, separadas pelo caracter ':' (dois-pontos), mas pode não o ser. Neste contexto, implemente três funções auxiliares:

- a) uma função `int lerLinha(char lin[])` que, fazendo uso da função `getchar()`, leia para o vector `lin` uma sequência de caracteres terminada pelo caracter mudança de linha (caracter que já não será guardado) ou por fim de input. No fim da string deve colocar o terminador convencional para strings. A função retornará o número de caracteres lidos para o vector. Admita que o vector tem espaço suficiente para guardar a sequência. Não poderá fazer uso de outras funções de sistema.
- b) uma função `int posicao(char lin[],char c)` que retorne o índice da posição da string `lin` que contém o caracter dado em `c`, se ocorrer uma só vez (se não, retorna -1).
- c) uma função `char analisaLinha(char lin[])` que analise a sequência dada por `lin` e retorne um único caracter com o seguinte significado:
- N O número de ocorrências do caracter dois-pontos na linha não é 1
  - E A parte esquerda (antes dos dois-pontos) é mais comprida que a da direita
  - D A parte direita (depois dos dois pontos) é mais comprida que a esquerda
  - C A parte esquerda e direita têm o mesmo comprimento, mas são diferentes
  - I A parte esquerda e direita são exactamente iguais.

Exemplos:

Entrada	Saída
-----	
Simples Simples	N
Curto:Comprido	D
Simples:Simples	I

3. Considere a representação de datas de acordo com a estrutura de dados seguinte:

```
typedef struct {
    int ano; // supomos ano >1900
    int mes; // mes de 1 a 12
    int dia; // dia de 1 a 31 (nº de dias depende do mes)
} DATA;
```

A título de exemplo, uma variável `d` que represente a data de 2011/01/28 (28 de Janeiro de 2011) pode ser declarada do seguinte modo:

```
DATA d= {2011,1,28};
```

Implemente as funções seguintes:

- a) `int compara(DATA d1, DATA d2)` para comparar duas datas `d1` e `d2`, retornando um inteiro negativo se `d1` precede `d2`, zero se forem iguais e um inteiro positivo se `d1` for posterior a `d2`.
- b) `DATA adicionaDias(DATA d, int nd)` para, dada uma data `d`, determinar a data que corresponde à data inicial `d` acrescida do número de dias dado por `nd`. Suponha já definida uma função `BOOL bissexto(int ano)` que retorna 1 se o ano dado em `ano` for bissexto e 0 caso contrário.
- c) `int maxAniversarios(DATA datas[],int n,int ano)` para analisar um vector que contém `n` datas de nascimento e retornar o número máximo de aniversários num mês, num certo `ano` dado por `ano`. Admita que, se o `ano` não for bissexto, os aniversários de 29 de Fevereiro transitam para 1 de Março.

Departamento de Ciência de Computadores  
**Introdução à Programação (CC111)**

FCUP  
**2010/11**

Exame: 28/Janeiro/2011

duração: 3 horas

4. Nas alíneas seguintes supõe-se que a matriz `char nomes[] [MAXN]` guardará nomes de alunos de um curso e que a matriz `int dnotas[] [7]` guardará as suas classificações a cinco disciplinas (A, B, C, D e E), a média arredondada dessas classificações e um desvio para tal média. O desvio é igual à soma dos quadrados das diferenças entre as notas e a média arredondada (note que, para arredondar a média, basta somar 0.5 ao valor antes de o guardar na matriz). Por exemplo, poderia ter:

```
nomes[0] [] = "Joao Diogo Matias"    dnotas[0] [] = {12,12,12,11,10,11,4}
nomes[1] [] = "Jorge Maria Faro"     dnotas[1] [] = {16,17,18,16,18,17,4}
nomes[2] [] = "Antonia Faria"        dnotas[2] [] = {16,14,14,17,12,15,16}
nomes[3] [] = "Maria Dinis"          dnotas[3] [] = {15,15,15,15,15,15,0}
...
```

- a) Escreva uma função `void dados(int na, char nomes[] [MAXN], int dnotas[] [7])` para ler da entrada padrão os dados sobre `na` alunos e guardar a informação em memória. Para cada aluno, é dado o nome numa linha e na linha seguinte cinco notas (de 10 a 20) separadas por espaços. Os valores da média arredondada e do desvio das classificações serão calculados e guardados também em `dnotas` (ficam nas últimas colunas da matriz).
- b) Escreva uma função `void nivelNota(int media, char nivel[])` em que dada a média de um aluno, `media`, devolve uma string `nivel` com o correspondente nível qualitativo da nota, dado por: 17 - 20 Muito Bom, 14 - 16 Bom, e 10 - 13 Suficiente.
- c) Escreva uma função para imprimir uma tabela a partir da informação existente, apresentando os alunos pela ordem passada num vector `int ordem[]`. Por exemplo, se esse vector tivesse 1,3,2,0, e se `na=4` e `MAXN=21`, o resultado seria:

Aluno	A	B	C	D	E	Media	Nivel
Jorge Maria Faro	16	17	18	16	18	17	Muito Bom
Maria Dinis	15	15	15	15	15	15	Bom
Antonia Faria	16	14	14	17	12	15	Bom
Joao Diogo Matias	12	12	12	11	10	11	Suficiente

- d) Apresente as ideias principais dos três métodos de ordenação dados na disciplina (selecção, inserção e bolha), destacando o que os distingue. Baseando-se num desses métodos, e na sequência da alínea anterior, implemente uma função que produza um vector `int ordem[]` para apresentação da tabela os alunos por ordem decrescente de média (arredondada), e, em caso de empate, por ordem crescente de desvio das classificações, e se persistir, pela ordem de ocorrência nas estruturas de dados guardadas. O número de alunos e os dados existentes são passados a essa função. **(FIM)**