



Nota	
------	--

Aluno: _____

Matrícula: _____ Turma: _____

Questão 1) (1,5 pontos)

- a) (1,0 pontos) Escreva uma função com nome **diaseguinte** que possui quatro parâmetros. Os três primeiros parâmetros são ponteiros para inteiros representando endereços de memória onde estão armazenados conteúdo de **dia**, **mês** e **ano**, respectivamente, para formar uma data. O quarto parâmetro é um vetor com 12 posições onde cada posição armazena a quantidade de dias dos meses do ano, isto é, na primeira posição está armazenado a quantidade de dias do mês de janeiro, na segunda posição a quantidade de dias do mês de fevereiro, e assim sucessivamente. A função **diaseguinte** deve modificar **dia**, **mês** e **ano** de forma que eles representem o dia seguinte.

Exemplo1

Entrada: **Dia:30 Mês:04 Ano:2010**
VDias: {31,28,31,30,31,30,31,31,30,31,30,31}
Saída: **Dia:01 Mês:05 Ano:2010**

Exemplo 2

Entrada: **Dia:31 Mês:12 Ano:2010**
VDias: {31,28,31,30,31,30,31,31,30,31,30,31}
Saída: **Dia:01 Mês:01 Ano:2011**

Exemplo3

Entrada: **Dia:29 Mês:04 Ano:2010**
VDias: {31,28,31,30,31,30,31,31,30,31,30,31}
Saída: **Dia:30 Mês:04 Ano:2010**

- b) (0,5 ponto) Complete o programa para fazer chamada da função **diaseguinte** preenchendo a linha pontilhada:

```
int main(void){
    int dia,mes,ano;
    int vdias[12] = {31,28,31,30,31,30,31,31,30,31,30,31};

    dia=30;
    mes= 4;
    ano=2010;

    /* chamada da função diaseguinte */
    .....

    printf("\n %d/%d/%d", dia,mes,ano);
    return 0;
}
```

Departamento de Informática - PUC-Rio
INF 1005 – Programação I
P2– 18/05/2011



Nota	
-------------	--

Aluno: _____

Matrícula: _____ Turma: _____



Aluno: _____

Matrícula: _____ Turma: _____

Questão 2) (4,5 pontos) Tentando descobrir se um dado de seis faces era viciado, um dono de cassino honesto lançou este dado várias vezes armazenando os resultados no arquivo chamado "lances.txt".

α) (2,0 ponto) Escreva uma função para contar as ocorrências de cada face. A função deve receber como parâmetros o número de elementos no vetor e o vetor de ocorrências de cada face zerado, retornando este vetor preenchido com a quantidade de ocorrências de cada face e o número de lançamentos realizados pelo dono do cassino. Esta função deve manipular o arquivo "lances.txt" para obter os resultados dos lançamentos. Os valores lidos do arquivo "lances.txt" variam de 1 a 6 e são sempre válidos. A função deve seguir o seguinte protótipo:

```
int conta_ocorrencias_faces (int n ,int vOcorFaces[]);
```

β) (1,5 ponto) Escreva uma função para mostrar na tela os números das faces viciadas ou a mensagem "Dado não viciado". A função deve receber como parâmetros o número de elementos no vetor, o vetor com os valores das ocorrências de cada face e o número de lançamentos realizados pelo dono do cassino. A função deve seguir o seguinte protótipo:

```
void exibir (int n ,float vOcorFaces[], int totlançamentos);
```

χ) (1,0 ponto) Escreva um programa em C (função main) que leia do arquivo "lances.txt" os resultados dos lançamentos e mostre, como saída, as faces viciadas ou a mensagem "Dado não viciado". Seu programa deverá obrigatoriamente usar as funções definidas nos itens a e b.

Observação: A face de um dado é considerada viciada se sua ocorrência for superior a 18% dos lançamentos ou inferior a 15% dos lançamentos.

Exemplo 1:

Entrada (arquivo "lances.txt"): 1 2 2 1 3 4 3 4 5 6 6 5

Saída: Dado não viciado

Exemplo 2:

Entrada (arquivo "lances.txt"): 1 1 2 2 2 3 3 3 4 4 3 4 5 5 6 5 5 6

Saída: Faces Viciadas: 3 5

Departamento de Informática - PUC-Rio
INF 1005 – Programação I
P2– 18/05/2011



Nota	
-------------	--

Aluno: _____

Matrícula: _____ Turma: _____



Aluno: _____

Matrícula: _____ Turma: _____

Questão 3) (4,0 pontos) As disciplinas (no máximo 20) de uma universidade possuem códigos que as identificam e critérios de avaliação escolhidos entre 5 critérios possíveis. Os critérios são numerados de 1 a 5 e diferem entre si no número de provas realizadas. As disciplinas são codificadas por um número qualquer de 2 algarismos.

Escreva um programa em C que armazene no arquivo "NumProvas.txt" o número de provas de cada uma das disciplinas nas quais um aluno está inscrito.

Para carregar as informações das disciplinas (código e critério) e o número de provas de cada critério de avaliação em vetores, seu programa deve **obrigatoriamente** usar as funções abaixo (que não devem ser implementadas por você):

```
int ler_disciplinas (int n, int vCodDisc[], int vCritDisc[]):  
que preenche os vetores recebidos com o código e critério utilizado pelas no máximo 20  
disciplinas
```

```
void ler_critérios (int n, int vProvasCrit[]);  
que preenche o vetor recebido com o número de provas de cada um dos 5 critérios por  
ordem de critério (na primeira posição armazena a quantidade de provas do critério 1, na  
segunda posição armazena a quantidade de provas do critério 2 e, assim, sucessivamente).
```

Após carregar os dados, seu programa deve ler do teclado o código de cada uma das disciplinas nas quais o aluno está inscrito, escrevendo no arquivo "NumProvas.txt", os dados desejados. Se o aluno fornecer um código de disciplina inválido a mensagem "Codigo de disciplina invalida" deve ser mostrada na tela. O aluno finaliza a entrada de dados digitando, como código da disciplina, o número 0 (zero).

Atenção:

- as funções *ler_disciplinas* e *ler_critérios* **não** devem ser **implementadas, apenas utilizadas**.
- além das duas funções auxiliares descritas acima, seu programa **deve** utilizar uma **função de busca** criada por você.

Exemplo:

Código da Disciplina	Critério
23	1
18	2
28	3
32	5
24	4
45	1
54	2
17	4

	Número de Provas por Critério
1º	3
2º	2
3º	5
4º	4
5º	6

Observe que o número de provas do critério *i* está indicado na *i*-ésima linha do vetor "numero de provas".

Por exemplo, na disciplina 32, como utiliza o critério 5, são aplicadas 6 provas.

Exemplo: Dados de entrada:

```
18 /*disciplina 1*/  
28 /*disciplina 2*/  
32 /*disciplina 3*/  
0 /*disciplina 4*/
```

Saída do arquivo "NumProvas.txt"

```
Disciplina 18: 2 provas  
Disciplina 28: 5 provas  
Disciplina 32: 6 provas
```



Aluno: _____

Matrícula: _____ Turma: _____

Cabeçalho das funções que podem ser úteis:

math.h

```
double exp(double x);  
double fabs (double x);  
double pow(double x, double y);  
double sqrt(double x );  
double sin (double x);  
double cos (double x);
```

stdio.h

```
int printf (char* format, ...);  
int scanf (char* format, ...);  
FILE* fopen (char* _lename, char* mode);  
int fclose (FILE* fp);  
int fprintf (FILE* fp, char* format, ...);  
int fscanf (FILE* fp, char* format, ...);  
int feof (FILE* fp);
```