



Departamento de Informática - PUC-Rio
INF 1007 – Programação 2
P1 – 17/04/2010



Aluno: _____

Matrícula: _____ Turma: _____

Instruções:

- 1) Escreva seu nome completo, matrícula e turma em todas as folhas desta prova;
- 2) A prova deve ser completamente resolvida nas folhas que constam deste caderno, utilizando-se frente e/ou verso;
- 3) As questões podem ser resolvidas em qualquer ordem;
- 4) As soluções que não forem apresentadas nas páginas a elas destinadas devem ser identificadas com o número da questão a que se referem;
- 5) A prova pode ser feita utilizando-se lápis ou caneta (azul ou preta);
- 6) Todos os dispositivos eletrônicos (celulares, i-pods, etc) devem ser desligados.

Pontuação:

Questão	Item	Valor	Nota
1	A	2,0	
	B	2,0	
2	A	1,0	
	B	2,0	
3	Único	3,0	
Total		10,0	

INF 1007 – P1 – 17/04/10	
Nome:	
Matrícula:	Turma:

Questão 1

A) [Valor: 2,0 pontos] Escreva uma função em C que determina a média e a situação de um aluno em uma disciplina. A função recebe como parâmetros as três notas de um aluno ($p1$, $p2$, e $p3$), seu número de faltas ($faltas$), o número total de aulas da disciplina ($aulas$) e o ponteiro para uma variável ($media$), conforme o seguinte protótipo:

```
char situacao(float p1, float p2, float p3, int faltas,
             int aulas, float *media);
```

Na variável indicada pelo ponteiro $media$, a função deve armazenar a média do aluno, calculada como a média aritmética das três provas. Além disso, a função deve retornar um caractere indicando a situação do aluno no curso, definido de acordo com o seguinte critério:

Número de Faltas	Média	Situação	Retorno
Menor ou igual a 25% do total de aulas	Maior ou igual 6,0	Aprovado	A
	Menor que 6,0	Reprovado	R
Maior que 25% do total de aulas	Qualquer	Reprovado por faltas	F

B) [Valor: 2,0 pontos] Escreva um programa completo em C que utiliza a função anterior para determinar a situação de um aluno. O programa deve:

- ler do teclado três números reais e dois números inteiros, representando as notas da $p1$, $p2$ e $p3$, o número de faltas e o número de aulas, respectivamente;
- **chamar a função desenvolvida na primeira questão** para determinar a média e a situação do aluno na disciplina;
- exibir a média (com apenas uma casa decimal) e a situação do aluno, isto é, “APROVADO”, “REPROVADO” ou “REPROVADO POR FALTAS”, dependendo do caractere retornado pela função, conforme a tabela acima.

Veja a seguir um exemplo da entrada e a respectiva saída esperada:

9.0 9.5 4.0 3 20
7.5 APROVADO

INF 1007 – P1 – 17/04/10	
Nome:	
Matrícula:	Turma:

Solução da Questão 1:

```
/* ITEM A */
```

```
char situacao(float p1, float p2, float p3, int faltas,
              int aulas, float *media)
{
    float percentual_faltas = faltas*100/(float)aulas;
    float med = (p1+p2+p3)/3;

    *media = med;

    /* mais que 25% de faltas */
    if (percentual_faltas > 25)
        return 'F';
    else
    {
        /* media maior ou igual 6,0 */
        if (med >= 6.0) return 'A';
        /* media menor 6,0 */
        else return 'R';
    }
}
```

```
/* ITEM B */
```

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

```
int main(void)
{
    float p1, p2, p3;
    int aulas, faltas;
    char sit;
    float media;

    scanf("%f %f %f %d %d", &p1, &p2, &p3, &faltas, &aulas);

    sit = situacao(p1, p2, p3, faltas, aulas, &media);

    if (sit == 'A') printf("\n%.1f APROVADO\n\n", media);
    else if (sit == 'R') printf("\n%.1f REPROVADO\n\n", media);
    else printf("\n%.1f REPROVADO POR FALTA\n\n", media);

    return 0;
}
```

Nome:

Matrícula:

Turma:

Questão 2

Considere que são utilizados dois vetores de inteiros para controlar a presença de alunos a uma aula. A turma tem N alunos, mas apenas K alunos compareceram a uma determinada aula. O vetor *turma*, de tamanho N , contém as matrículas de todos os alunos da turma, armazenadas em ordem aleatória. O vetor *presentes*, de tamanho K , é utilizado para armazenar somente as matrículas dos alunos que compareceram a essa aula, também armazenadas em ordem aleatória.

A) [Valor: 1,0 ponto] Escreva uma função em C que verifica se um determinado aluno compareceu à aula. A função recebe como parâmetros o ponteiro *presentes*, para o vetor de inteiros com as matrículas dos alunos presentes, o tamanho k desse vetor e o inteiro *aluno*, com matrícula do aluno a ser procurada. A função deve retornar 1, se a matrícula do aluno aparece no vetor de presentes, ou 0, se não for encontrada. A função tem o seguinte protótipo:

```
int buscaAluno(int *presentes, int k, int aluno);
```

B) [Valor: 2,0 pontos] Escreva uma função em C que crie um novo vetor contendo as matrículas dos alunos que faltaram à aula. Essa função:

- recebe como parâmetros o ponteiro *presentes* para o vetor de inteiros com as matrículas dos alunos presentes, o tamanho k desse vetor, o ponteiro *turma* para o vetor de inteiros com as matrículas de todos os alunos da turma e o tamanho n desse vetor;
- utiliza a função *buscaAluno*, implementada no item anterior, para verificar se cada aluno compareceu à aula. Se não compareceu, sua matrícula deve ser armazenada em um novo vetor de inteiros com as matrículas dos alunos faltosos;
- retorna o ponteiro para o novo vetor de inteiros contendo apenas as matrículas dos alunos faltosos, que deve ser alocado dinamicamente **exatamente com o tamanho necessário** para armazenar essa informação. Se não houver faltas ou não for possível alocar dinamicamente o novo vetor, a função deve retornar NULL.

Por exemplo, se o vetor de presentes contiver os valores {6, 1, 9, 7, 12, 3, 4}, com k igual a 7 e o vetor da turma contiver os valores {10, 2, 7, 4, 11, 6, 3, 8, 9, 1, 5, 12}, com n igual 12, o vetor retornado deverá conter os valores {10, 2, 11, 8, 5}. A função tem o seguinte protótipo:

```
int* faltosos(int *presentes, int k, int *turma, int n);
```

INF 1007 – P1 – 17/04/10	
Nome:	
Matrícula:	Turma:

Solução da Questão 2:

```
/* ITEM A */
```

```
int buscaAluno(int *presentes, int k, int aluno)
{
    int i;

    for (i = 0; i < k; i++)
        if (presentes[i] == aluno)
            return 1;

    return 0;
}
```

```
/* ITEM B */
```

```
int* faltosos(int *presentes, int k, int *turma, int n)
{
    int i, j, resp;

    int *vet_faltas;

    if (n == k) return NULL;

    vet_faltas = (int*) malloc (sizeof(int)*(n-k));

    if (vet_faltas == NULL) return NULL;

    for(i=0, j=0; i < n; i++)
    {
        resp = buscaAluno(presentes, k, turma[i]);

        if (resp == 0)
        {
            vet_faltas[j] = turma[i];
            j++;
        }
    }
    return vet_faltas;
}
```

Questão 3

[Valor: 3,0 pontos] Escreva uma função em C que receba como parâmetro uma cadeia de caracteres, **contendo apenas letras entre 'A' e 'Z' ou 'a' e 'z' e espaços em branco**, e retorne uma nova cadeia -- alocada dinamicamente --, contendo uma cópia da cadeia original, mas sem espaços em branco no início ou no final, isto é, sem qualquer caractere ' ' (espaço) localizado antes da primeira letra ou depois da última letra da cadeia. Por simplificação, considere que a cadeia de entrada possui no mínimo uma letra (entre 'A' e 'Z' ou 'a' e 'z'). Por exemplo, se a cadeia “ nota do aluno ” (com três espaços em branco no início e dois no final) for passada para a função, a cadeia retornada deve ser “nota do aluno”, sem nenhum espaço no início ou no final. Se não for possível alocar dinamicamente a nova cadeia, a função deve retornar NULL. O protótipo da função é:

```
char* compactar(char *s);
```

Solução da Questão 3:

```
char *compactar(char *s)
{
    int i, j, ini, fim;
    char *nova;

    nova = (char*) malloc (sizeof(char)*(strlen(s)+1));

    if (nova == NULL) return NULL;

    /* posicao da primeira letra da cadeia */
    for (ini=0; s[ini] == ' '; ini++);

    /* posicao da ultima letra da cadeia */
    for (fim=strlen(s)-1; s[fim] == ' '; fim--);

    /* copia caracteres para nova cadeia */
    for (i=ini, j=0; i<=fim; i++, j++)
        nova[j] = s[i];

    /* termina nova cadeia */
    nova[j] = '\0';

    return nova;
}
```

Protótipos de funções que podem ser úteis:

stdio.h:

```
int scanf (char* formato, ...);
int printf (char* formato, ...);
FILE* fopen (char* nome, char* modo);
int fclose (FILE* fp);
int fscanf (FILE* fp, char* formato, ...);
int fprintf (FILE* fp, char* formato, ...);
char* fgets(char* str, int size, FILE* fp);
int sscanf(char* str, char* formato, ...);
```

math.h:

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

string.h:

```
int strlen (char* s);
int strcmp (char* s, char *t);
char* strcpy (char* destino, char* fonte);
char* strcat (char* destino, char* fonte);
```

stdlib.h:

```
void* malloc (int nbytes);
void free (void* p);
void qsort (void *vet, int n, int tam, int (*comp) (const void*, const void*));
```