

INF 1620 – P1 - 13/09/08	Questão 1
Nome:	
Matrícula:	Turma

[Valor: 3,0 pontos] Para um conjunto de n resistores dispostos em paralelo, a resistência equivalente R_{eq} é inversamente proporcional ao somatório do inverso das resistências, ou seja:

$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots + \frac{1}{R_n}$$

Essa fórmula é válida somente quando os valores das n resistências são não-nulos. Se uma das resistências for nula, a resistência equivalente R_{eq} também será nula, independentemente do valor das demais.

Escreva uma função em C que calcule o valor da resistência equivalente a um conjunto de resistores ligados em paralelo. Conforme o protótipo abaixo, esta função recebe como parâmetros um vetor de n números reais contendo os valores das resistências (representado pelo inteiro n que indica o tamanho do vetor e o ponteiro res para seu primeiro elemento), e deve retornar o valor da resistência equivalente.

```
float req(int n, float *res);
```

Obs.: A fim de facilitar o teste da função implementada, fornecemos o programa completo transcrito a seguir, onde é executado um único caso de teste.

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

#define N 3

/* código da sua funcao*/

int main(void)
{
    float f[N] = {3.2, 4.4, 6.7};
    float r;
    printf("Executando um caso de teste:\n");
    r = req(N , f);
    printf("Solucao Esperada: 1.451\n");
    printf("Solucao Obtida:   %.3f\n", r);
    return 0;
}
```

Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.

INF 1620 – P1 - 13/09/08	Questão 2
Nome:	
Matrícula:	Turma

[Valor: 3,0 pontos] Considere que você armazenou os valores observados em um experimento de física em um vetor de números reais (do tipo float). Para melhorar a precisão dos seus dados, você deve eliminar o maior e o menor valor dentre todas as medições. Por exemplo, para o vetor (1.2, 1.2, 0.9, 1.5, 1.4, 1.5), o terceiro elemento deveria ser retirado por ser o de menor valor, e também o quarto (ou o sexto elemento, a escolha é indiferente) por ser o de maior valor.

Escreva uma função em C que receba como parâmetros um vetor (*vet*) de n números reais (assuma que n é sempre maior do que 3), correspondendo às medições de um experimento, e retorne um novo vetor, alocado dinamicamente, sem os elementos do vetor original que têm o maior e o menor valor. Se não for possível alocar o novo vetor, a função deve retornar NULL.

```
float* medicao(int n, float *vet);
```

Obs.: A fim de facilitar o teste da função implementada, fornecemos o programa completo transcrito a seguir, onde é executado um único caso de teste. Neste programa só são comparados os primeiros e os últimos elementos de cada vetor. Para ter certeza de que sua solução está correta, o aluno deve alterar esse programa a fim de imprimir e comparar os dois vetores completos.

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

#define N 6

/* código da sua funcao*/

int main(void)
{
    float a[N] = {5.3, 7.1, 5.8, 4.9, 6.8, 9.2};
    float *b;
    printf("Executando um caso de teste:\n");
    b = medicao(N, a);
    printf("Primeiro elemento do vetor original: %.1f\n", a[0]);
    printf("Primeiro elemento do novo vetor:      %.1f\n", b[0]);
    printf("Ultimo elemento do vetor original:    %.1f\n", a[N-1]);
    printf("Ultimo elemento do novo vetor:        %.1f\n", b[N-3]);

    /* Procure imprimir aqui todos os elementos desses vetores*/

    free(b);
    return 0;
}
```

Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.

INF 1620 – P1 - 13/09/08	Questão 3
Nome:	
Matrícula:	Turma

A política de segurança de uma empresa exige que cada usuário modifique periodicamente sua senha de acesso ao sistema. Essa senha deve ter no mínimo 8 caracteres, no máximo 50 caracteres e não pode conter o caracter de espaço em branco. Além disso, na troca de senha, a nova senha não pode conter mais do que quatro caracteres idênticos aos da senha antiga na mesma posição. Por exemplo, se a senha antiga é “asf35tWh”, a senha nova pode ser “ASF715tWh”, já que todas as posições das senhas têm valores diferentes, mas não poderia ser “asf67tW8yha”, pois as letras “a”, “s”, “f”, “t” e “W” aparecem na mesma posição em ambas as senhas. Observe que letras maiúsculas e minúsculas são tratadas como caracteres distintos.

a) [Valor: 2,0 pontos] Escreva uma função em C que receba como parâmetros duas *strings* *s1* e *s2*, contendo a senha antiga e a nova, respectivamente, e retorne 0 se a senha nova for válida, ou 1, se for inválida. A função deve ter o seguinte protótipo:

```
int valida_senha (char *s1, char *s2);
```

b) [Valor: 2,0 pontos] Escreva um programa completo em C para testar a função desenvolvida no item anterior. Este programa deverá:

1. Escrever uma mensagem solicitando que o usuário digite uma nova senha;
2. Ler do teclado uma *string* fornecida pelo usuário;
3. Verificar se essa *string* tem o tamanho adequado, caso contrário o programa deve imprimir a mensagem “Senha de tamanho inadequado, tente outra vez” e retornar ao passo 1, solicitando que o usuário digite uma nova senha.
4. Comparar, utilizando a função *valida_senha*, a *string* fornecida pelo usuário com a *string* “asf35tWh”, que representa a senha antiga.
5. Se a *string* fornecida corresponder a uma nova senha válida, o programa deve imprimir a mensagem “A senha foi atualizada” e terminar sua execução.
6. Por outro lado, se a *string* fornecida não for uma senha válida, o programa deve imprimir a mensagem “Nova senha invalida, tente outra vez” e retornar ao passo 1, solicitando que o usuário digite uma nova senha.

Veja a seguir um exemplo da saída esperada:

```
Digite uma nova senha:
ASF71@t34h736rsfsatryq763h6F
Senha de tamanho inadequado, tente outra vez

Digite uma nova senha:
asf67tW8yha
Senha invalida, tente outra vez

Digite uma nova senha:
ASF715tWh
A senha foi atualizada
```

Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.

RASCUNHO

Respostas nesta folha não serão consideradas.

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* origem);
char* strcat (char* destino, char* origem);
```

stdlib.h:

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

Não separe as folhas deste caderno.