

INF 1620 – P4 - 01/07/08	Questão 1
Nome:	
Matrícula:	Turma

Escreva uma função em C que receba duas strings, cujos caracteres estão ordenados em ordem crescente, e retorne uma nova string alocada dinamicamente, que contém todos os caracteres das strings originais. A string retornada também deve estar ordenado em ordem crescente com repetição de valores.

Por exemplo, seja $S1 = \text{"abefk"}$ e $S2 = \text{"acit"}$ a string resultante será "aabcefikt"

O protótipo da função deve ser:

```
char* merge (char* S1, char* S2);
```

Observação: As cadeias $S1$ e $S2$ não serão vazias.

INF 1620 – P4 - 01/07/08	Questão 2
Nome:	
Matrícula:	Turma

Ao final de um campeonato as informações sobre o número de vitórias, empates e derrotas de cada time estão armazenadas em um vetor de ponteiros para a estrutura abaixo:

```
struct time{
    char Nome[61];
    int Vitorias;
    int Empates;
};
```

```
typedef struct time Time;
```

Sabendo que a pontuação é calculada considerando que cada vitória vale 3 pontos e o empate 1 ponto, utilizando uma das técnicas de ordenação vistas no curso, escreva uma função para colocar o vetor de ponteiros para a estrutura Time em ordem crescente de pontuação (o time de menor pontuação deve ser o primeiro). A função deve receber como parâmetros o vetor que armazena ponteiros para a estrutura do tipo **Time** e o número de times, de acordo com o protótipo definido a seguir:

```
void Ordena_Times(Time** Campeonato, int n);
```

INF 1620 – P4 - 01/07/08	Questão 3
Nome:	
Matrícula:	Turma

Dada uma lista encadeada que armazena a média dos alunos de uma turma, cujo tipo que representa um nó da lista é dado por:

```
struct tipo_al {
    int Mat;
    char Nome[81];
    float Media;
};
typedef struct tipo_al Aluno;

struct Lista {
    Aluno info;
    struct lista *prox;
};
typedef struct lista Lista;
```

Observe que cada nó da lista possui como informação a estrutura correspondendo aos dados alunos. Implemente uma função que receba uma lista encadeada do tipo *Lista* e retorne uma nova lista dos alunos aprovados, ou seja, os alunos com média ≥ 7.0

O protótipo da função é dado por:

```
Lista* Copia_Aprovados (Lista* L);
```

Observação: a Lista L deverá permanecer inalterada.

INF 1620 – P4 - 01/07/08	Questão 4
Nome:	
Matrícula:	Turma

Uma árvore binária de busca é definida com a propriedade de que em qualquer nó n , todos os elementos da sub-árvore esquerda de n têm valores menores que o de n , e todos os elementos da sub-árvore direita têm valores maiores ou iguais que o de n .

Dado o tipo correspondente:

```
typedef struct no ARV;  
struct no {  
    int val;  
    struct no *esq, *dir;  
};
```

Escreva uma função com protótipo:

```
ARV* Busca (ARV* a, int x);
```

que retorne o endereço do nó da árvore (ponteiro para o nó da árvore) com o valor desejado (x). Caso este valor não esteja na árvore, deve ser retornado o endereço do nó imediatamente sucessor, isto é, o nó que possui o valor imediatamente superior. No caso da árvore ser vazia ou de não haver um sucessor (se x for maior que o maior valor armazenado na árvore) deverá ser retornado o endereço NULO (NULL).

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* 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*));
```

Não separe as folhas deste caderno.