

INF 1620 – P3 - 07/12/01	Questão 1
Nome:	
Matrícula:	Turma

Considere a implementação de conjuntos de números reais usando listas encadeadas. Nestes conjuntos, os elementos (no caso, números reais) são armazenados em ordem crescente. O tipo da linguagem C que representa estes conjuntos (SET) é dado por:

```
struct no {
    float valor;          /* valor do elemento */
    struct no* prox;
};
typedef struct no *SET;
```

Escreva uma função que retorne o número de elementos da interseção de dois conjuntos dados, isto é, a função deve retornar o número de elementos comuns aos dois conjuntos. O protótipo da função deve ser:

```
int inter (SET s1, SET s2);
```

Obs: não é necessário criar o conjunto resultante da interseção dos dois conjuntos, apenas determinar o número de elementos que existiriam na interseção.

INF 1620 – P3 - 07/12/01	Questão 2
Nome:	
Matrícula:	Turma

Conjuntos de inteiros que variam de 0 (zero) a 255 podem ser implementados com vetores de bits usando a estrutura de dados definida pelo tipo

```
typedef int BITSET[8];
```

desde que o tipo `int` tenha 4 bytes. Implemente duas funções `insere` e `diferenca` cujos protótipos são, respectivamente:

```
void insere(BITSET s, int i);
```

```
void diferenca(BITSET s1, BITSET s2, BITSET s3);
```

A função `insere` inclui o elemento `i` no conjunto `s`.

A função `diferenca` calcula a diferença entre os conjuntos `s2` e `s3`, que passa a ser o novo valor de `s1`. Ou seja, na saída da função `diferenca`, o conjunto `s1` contém os elementos que estavam em `s2`, mas que *não* estavam no conjunto `s3`.

Alguns operadores úteis de C:

| ou bit a bit (bitwise or)

& e bit a bit (bitwise and)

^ ou exclusivo (bitwise exclusive or)

~ negação bit a bit (bitwise negation)

<< deslocamento para esquerda (left shift)

>> deslocamento para direita (right shift)

% resto da divisão

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 – P3 - 07/12/01	Questão 3
Nome:	
Matrícula:	Turma

Considere uma matriz de dimensão $m \times n$, onde m representa o número de linhas da matriz e n o número de colunas. Esta matriz pode ser representada por um vetor de dimensão $m*n$, onde os elementos da primeira linha da matriz ocupam as primeiras posições do vetor, seguidos dos elementos da segunda linha, e assim por diante, conforme ilustrado esquematicamente abaixo:

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{bmatrix} \longrightarrow [a \ b \ c \ d \ e \ f \ g \ h \ i \ j \ k \ l]$$

Supondo que matrizes $m \times n$ de números reais são armazenadas em vetores, escreva uma função que, dada uma matriz, crie dinamicamente e retorne a sua transposta (de dimensão $n \times m$). O protótipo da função deve ser:

```
float* transposta (int m, int n, float* mat);
```

Notas:

- Se Q é a matriz transposta de M , tem-se que $Q[i][j]$ é igual a $M[j][i]$.
- O resultado desta função independe da convenção de numeração das linhas e colunas da matriz. O aluno pode optar por numerar a partir de zero ou um.

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 – P3 - 07/12/01	Questão 4
Nome:	
Matrícula:	Turma

Considere um arquivo texto “entrada.txt” que armazena as notas dos alunos. Cada linha do arquivo armazena duas informações: a matrícula do aluno, representada por um número inteiro, e a nota do aluno, representada por um número real (float).

Escreva um programa completo cuja entrada é o arquivo “entrada.txt” do diretório corrente, e cuja saída é um arquivo “saida.txt”, também no diretório corrente. O arquivo de saída difere do arquivo de entrada apenas por ter as informações por linha com ordem trocada: primeiro a nota depois a matrícula.

Por exemplo, se o arquivo de entrada contém as três linhas

```
9911234 7.6
9821234 6.3
9926789 8.3
```

O arquivo de saída correspondente deve conter

```
7.6 9911234
6.3 9821234
8.3 9926789
```

Alguns protótipos de funções úteis:

```
FILE *fopen(char *filename, char *mode );
int fclose( FILE *stream );
int fprintf( FILE *stream, char *format [, argument ]...);
int fscanf( FILE *stream, char *format [, argument ]... );
char *fgets( char *string, int n, FILE *stream );
```

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 – P3 - 07/12/01	Questão 5
Nome:	
Matrícula:	Turma

Considere um cadastro de alunos armazenado numa tabela *hash* que usa como chave de busca o número de matrícula. Considere também a existência de uma função de *hash* que, dado o número de matrícula, retorna o índice na tabela. O protótipo desta função é dada por:

```
int hash (int mat);
```

O tratamento de colisão é feito reservando-se parte da tabela para armazenar as colisões e formando-se um encadeamento dos elementos que colidiram. Assim, cada elemento guarda um índice para a posição da tabela onde está armazenado o elemento que colidiu. Um valor de índice igual a zero indica que não há mais elementos no encadeamento.

Suponha as seguintes declarações, relativas à tabela de hash dos alunos.

```
struct info {
    int matric;      /* número de matrícula do aluno */
    char nome[81];  /* nome do aluno */
    int prox;       /* índice do próximo elemento */
};
typedef struct info *PINFO;
typedef PINFO HASH[512];
```

Considerando que posições não preenchidas da tabela de *hash* armazenam o valor NULL, escreva uma função de busca que verifica se um aluno, dado seu número de matrícula, está armazenado no cadastro. A função deve retornar 1 se o aluno estiver no cadastro e 0 se não estiver. O protótipo da função deve ser:

```
int busca (HASH tab, int mat);
```

RASCUNHO

Não separe as folhas deste caderno. Respostas nesta folha não serão consideradas.