

INF 1620 – P2 - 23/10/04	Questão 1
Nome:	
Matrícula:	Turma

Considere um tipo abstrato de dados para representar uma disciplina da PUC-Rio, com as seguintes informações:

- Nome da disciplina: com até 50 caracteres
- Código da disciplina: com até 7 caracteres
- Quantidade de créditos: valor inteiro
- Lista encadeada com os alunos matriculados

Onde cada aluno é descrito pelas seguintes informações:

- Nome: com até 80 caracteres
- Matrícula: com até 7 caracteres
- Média na disciplina: número de ponto flutuante

- (a) Defina as estruturas em C, denominadas `disciplina` e `aluno`, que tenham os campos apropriados para guardar as informações de uma disciplina e seus alunos, conforme descrito acima.
- (b) Escreva uma função que receba como parâmetro um ponteiro para uma estrutura do tipo `disciplina`, definida no item anterior, e imprima na tela do computador uma linha com o código da disciplina e sua média geral (média das médias dos alunos). Essa função deve seguir o seguinte protótipo:

```
void imprime_media (struct disciplina* d);
```

INF 1620 – P2 - 23/10/04	Questão 2
Nome:	
Matrícula:	Turma

Considere a implementação de uma lista encadeada para armazenar números reais dada pelo tipo abaixo:

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

Implemente uma função que, dados uma lista encadeada e um número inteiro não negativo n , remova da lista seus n primeiros nós e retorne a lista resultante. Caso n seja maior do que o comprimento da lista, todos os seus elementos devem ser removidos e o resultado da função deve ser uma lista vazia. Essa função deve obedecer o seguinte protótipo:

```
Lista* retira_prefixo (Lista* l, int n);
```

INF 1620 – P2 - 23/10/04	Questão 3
Nome:	
Matrícula:	Turma

Considere a existência de um tipo abstrato `Fila` de números reais, cuja interface é definida pelo arquivo `fila.h` com o seguinte conteúdo:

```
typedef struct fila Fila;

Fila* cria (void);           /* cria uma fila vazia */
void insere (Fila* f, float v); /* insere um elemento */
float retira (Fila * f);     /* retira um elemento */
int vazia (Fila * f);       /* retorna se está vazia */
void libera (Fila * f);     /* libera a estrutura */
```

Sem conhecer a representação interna deste tipo abstrato e usando apenas as funções declaradas no arquivo `fila.h`, implemente uma função que, dada uma fila, inverta a ordem de seus elementos. Por exemplo, se a fila originalmente tiver seus elementos na ordem `4.8, 3.2, 9.8, 3.3`, após a chamada da sua função a fila deve ter seus elementos na ordem `3.3, 9.8, 3.2, 4.8`. O protótipo da função deve ser:

```
void inverte (Fila* f);
```

Dica: use uma pilha auxiliar para inverter os valores da fila ou faça uma implementação recursiva. Para usar uma pilha auxiliar, considere as operações de pilha definidas na última folha da prova.

INF 1620 – P2 - 23/10/04	Questão 4
Nome:	
Matrícula:	Turma

Considere uma árvore binária 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 ao de n .

Os tipos correspondentes são

```
struct arv {
    int info;
    struct arv *esq, *dir;
};
typedef struct arv Arv;
```

Escreva uma função com o protótipo

```
void imprime (Arv* a);
```

que imprime os valores dos nós da árvore de forma que este estejam ordena dos.

Por exemplo, dada a árvore:

```

graph TD
    6[6] --> 2[2]
    6 --> 8[8]
    2 --> 1[1]
    2 --> 4[4]
    4 --> 3[3]
    1 --- G1[ ]
    1 --- G2[ ]
    3 --- G3[ ]
    3 --- G4[ ]
    8 --- G5[ ]
    8 --- G6[ ]
  
```

Após a chamada da função estará impresso na tela a seguinte sequência:

1 2 3 4 6 8

Dica: Percorra a árvore em ordem simétrica.

INF 1620 – P2 - 23/10/04	Questão 5
Nome:	
Matrícula:	Turma

Numa estrutura de árvore, o nível (ou altura) de um nó é definido como sendo igual à distância da raiz até o nó em questão. Assim, o nó raiz está no nível 0, os filhos da raiz no nível 1, os netos no nível 2, e assim por diante.

Considerando uma estrutura de árvore binária definida pelo tipo abaixo:

```
struct arv {
    int info;
    struct arv *esq, *dir;
};
typedef struct arv Arv;
```

Escreva uma função que retorne a quantidade de nós em uma árvore até um nível n dado. O protótipo da função deve ser:

```
int conta (Arv* a, int n);
```

Dica: modifique o algoritmo de caminhamento em pré-ordem.

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);
```

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*));
```

pilha.h: (para a terceira questão da prova)

```
typedef struct pilha Pilha;

Pilha* cria (void); /* cria uma pilha vazia */
void push (Pilha* p, float v); /* insere um elemento no topo */
float pop (Pilha* p); /* retira o elemento do topo */
int vazia (Pilha* p); /* retorna se está vazia */
void libera (Pilha* p); /* libera a pilha */
```

Não separe as folhas deste caderno.