INF 1620 – P2 - 17/05/02	Questão 1
Nome:	
Matrícula:	Turma

Considere uma lista encadeada que armazena as notas de um conjunto de alunos. Os tipos dos dados são apresentados abaixo:

```
struct aluno {
   char nome[81];    /* nome do aluno */
   char mat[11];     /* matrícula do aluno */
   float nota;       /* nota do aluno */
};
typedef struct aluno Aluno;

struct no {
   Aluno info;
   struct no *prox;
};
typedef struct no No;
```

Escreva uma função que imprima o nome e a matrícula do aluno que tem a maior nota. A função recebe como parâmetro o ponteiro para o primeiro nó da lista e tem o seguinte protótipo:

```
void imprime (No* prim);
```

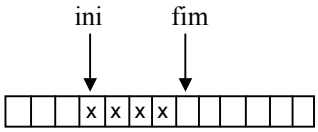
```
void imprime (No* prim)
{
   float nmax = 0.0;  /* armazena nota maxima */
   No* pmax = NULL;   /* ponteiro p/ aluno com nota maxima */
   No * p;

   for (p=prim; p!=NULL; p=p->prox) {
      if (p->info.nota > nmax) {
         nmax = p->info.nota;
         pmax = p;
      }
   }

   if (pmax != NULL) { /* teste opcional! */
      printf("%s\n", pmax->info.nome);    /* imprime nome */
      printf("%s\n", pmax->info.mat);     /* imprime matricula */
   }
}
```

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 – P2 - 17/05/02	Questão 2
Nome:	
Matrícula:	Turma

Considere uma fila dupla de valores reais que usa um vetor para armazenar seus elementos. O tipo que representa a fila é dado abaixo, onde `ini` indica o índice do vetor onde o elemento do início da fila está armazenado e `fim` indica o índice imediatamente após o último elemento da fila.

```
#define N 50
struct fila {
   int ini, fim;
   float vet[N];
};
typedef struct fila Fila;
```



Considerando a utilização de incremento e decremento circulares, pede-se:

(a) Escreva uma função para retirar um elemento do início da fila.

```
float retira_ini (Fila* f);
```

(b) Escreva uma função para retirar um elemento do fim da fila.

```
float retira_fim (Fila* f);
```

Notas: (a) Ambas as funções devem ter como valor de retorno o valor do elemento retirado da fila; (b) Deve-se considerar que a fila nunca estará vazia quando da chamada de uma dessas funções.

```
float retira_ini (Fila* f)
{
   float v = f->vet[f->ini];  /* armazena valor de retorno */
   f->ini = (f->ini+1)%N;     /* incremento circular */
   return v;
}

float retira_fim (Fila* f)
{
   int i;
   float v;

   /* decremento circular */
   i = f->fim - 1;
   if (i < 0)
      i = N-1;

   /* armazena valor de retorno */
   v = f->vet[i];

   /* atualiza indice da fila */
   f->fim = i;

   return v;
}
```

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

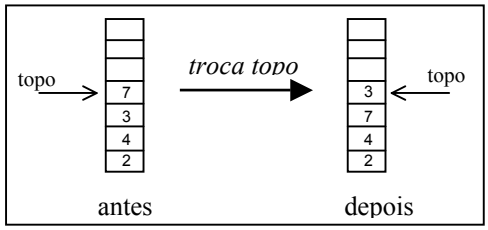
INF 1620 – P2 - 17/05/02	Questão 3
Nome:	
Matrícula:	Turma

Considere a implementação de uma pilha para armazenar valores inteiros implementada com o uso de lista encadeada. O tipo que representa a pilha é dado por:

```
struct no {
   int info;
   struct no* prox;
};
typedef struct no No;

struct pilha {
   No* prim;          /* aponta para o topo da pilha */
};
typedef struct pilha Pilha;
```

Escreva uma função que troque o valor do topo com o valor imediatamente "abaixo" do topo. O resultado dessa operação é ilustrado na figura ao lado. Essa função deve ter o seguinte protótipo:

```
Pilha* troca_topo (Pilha* p);
```



Nota: Considera-se que nunca haverá menos do que dois elementos na pilha.

```
/* Primeira solução: usar funções push e pop como auxiliaries */

void push (Pilha* p, int v)
{
   No* n = (No*)malloc(sizeof(No));
   n->info = v;
   n->prox = p->prim;
   p->prim = n;
}

int pop (Pilha* p)
{
   No* topo = p->prim;
   int v = topo->info;
   p->prim = topo->prox;
   free(topo);
   return v;
}

Pilha* troca_topo (Pilha* p)
{
   int v1 = pop(p);
   int v2 = pop(p);
   push(p,v1);
   push(p,v2);
   return p;
}
```

```
/* Segunda solução: fazer troca
interna sem funções auxiliaries */
Pilha* troca_topo (Pilha *p)
{
   No* topo1 = p->prim;
   No* topo2 = topo1->prox;
   topo1->prox = topo2->prox;
   topo2->prox = topo1;
   p->prim = topo2;
   return p;
}
/* Terceira solução: troca apenas as
informacoes */
Pilha* troca_topo (Pilha *p)
{
   int t = p->prim->info;
   p->prim->info =
          p->prim->prox->info;
   p->prim->prox->info = t;
   return p;
}
```

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 – P2 - 17/05/02	Questão 4
Nome:	
Matrícula:	Turma

Considerando a estrutura de árvore binária de valores reais definida pelo tipo abaixo:

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

Escreva uma função que retorne o número de nós da árvore cujo valor armazenado no campo `info` seja maior que um dado valor `x`. O protótipo da função é dado por:

```
int conta_maiores (Arv* a, float x);
```

Nota: a árvore vazia é representada pelo ponteiro nulo (`NULL`).

```
int conta_maiores (Arv* a, float x)
{

   if (a==NULL)    /* trata arvore vazia */
      return 0;

   else {          /* trata arvore nao vazia */
     int n;
     if (a->info > x)
       n = 1;
     else
       n = 0;

     return n + conta_maiores(a->esq,x) + conta_maiores(a->dir,x);
   }
}
```

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 – P2 - 17/05/02	Questão 5
Nome:	
Matrícula:	Turma

Numa estrutura de árvore, o nível de um nó é definido como sendo igual à distância do caminho 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.
Suponha as seguintes declarações, relativas a um tipo "árvore genérica", ou seja uma árvore cujos nós têm um número não limitado de filhos, que armazena valores inteiros.

```
struct arvgen {
   int info;
   struct arvgen *prim;  /* ponteiro p/ primeiro filho */
   struct arvgen *prox;  /* ponteiro p/ irmão */
};
typedef struct arvgen ArvGen;
```

Escreva uma função que retorne o nível do nó cuja informação seja igual a um valor `x` dado. Se não existir um nó com o valor `x`, a função deve ter como valor de retorno o valor `-1`. O protótipo da função é dado por:

```
int nivel (ArvGen* a, int x);
```

```
int nivel (ArvGen* a, int x)
{
   if (a->info == x)
      return 0;
   else {
      ArvGen* p;
      for (p=a->prim; p!=NULL; p=p->prox) {
         int n = nivel(p,x);
         if (n>=0)
            return 1+n;
      }
      return -1;
   }
}
```

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.

Não separe as folhas deste caderno.