MAC 115 – Introdução à Computação para
Ciências Exatas e Tecnologia
ESCOLA POLITÉCNICA – PRIMEIRO SEMESTRE DE 1998
Segunda Prova – 6 de junho de 1998

Nome do aluno: ______________________________

Assinatura: ______________________________

Nº USP: ______________ Turma: ______________

Professor: ______________________________

Instruções:

1. Não destaque as folhas deste caderno.
2. Preencha o cabeçalho acima.
3. A prova pode ser feita a lápis. Cuidado com a legibilidade.
4. A prova consta de 3 questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é permitido a consulta a livros, apontamentos ou colegas.
7. Considere nesta prova que o maior inteiro representável em C é estupidamente maior que 32767, ou seja, não utilize esse nem nenhum outro limite em seus programas.
8. Não é necessário apagar rascunhos no caderno de questões.

DURAÇÃO DA PROVA: 3 horas

Questão	Valor	Nota
1	2.0	
2	1.0	
	1.5	
	2.0	
3	1.5	
	1.0	
	1.0	
Total	10.0	

1. (valor 2.0 pontos)
   Simule a execução do programa abaixo, destacando a saída do programa. Use como entrada os seguintes dados: 5 1
   99 95 18 6 12

```
#include <stdio.h>
void imprima (int v[], int n){
  int i;
  for (i = 0 ; i < n ; i++) printf("%d ", v[i]);
  printf("\n");
}

void leia (int v[], int n){
  int i;
  for (i = 0; i < n ; i++) scanf("%d", &v[i]);
}

void troca (int *px, int *py){
  int aux;
  aux = *px;  *px = *py;   *py = aux;
}

void misterio (int v[], int n, int *p, int *r) {
  int aux, i, j;
  i = *p;  j = *r;  aux = v[(i + j)/2];
  while (i < j) {
     while (v[i] < aux ) i++;
     while (aux  < v[j]) j--;
     if (i <= j) {
        troca (&v[i] , &v[j]);
        i++; j--;
     }
     imprima(v, n);
  }
  *p = i;  *r = j;
}

void main(void){
  int seq[10], n, k, i, j, p, r;
  scanf("%d %d", &n, &k);
  printf("n = %d  k = %d\n", n, k);
  leia (seq, n); imprima (seq, n);
  p = 0;  r = n - 1;
  while (p < r) {
    i = p; j = r;
    misterio(seq, n, &i, &j);
    if (j <=  k) p = i ;
    if (k <=  i) r = j ;
    printf ("i = %d  j = %d p = %d r = %d\n", i, j, p, r);
  }
  imprima (seq, n);
  printf("Elemento = %d\n", seq[p]);
}
```

Para efeito de correção só será considerada a saída do programa. Você pode usar as duas tabelas abaixo como bem entender. Cada turma esta habituada a simular de maneira diferente, fazendo tabelas com "caras" diferentes das abaixo.

main						
seq	n	k	i	j	p	r

misterio						
v	n	p	r	aux	i	j

saída

2. (valor 1.0 + 1.5 + 2.0 = 4.5 pontos)
   Nesta questão você fará um programa em `C` que leia:

   - uma string `T`;
   - uma string `s1`; e
   - uma string `s2`,

   e que troca todas as ocorrências da string `s1` no texto `T` (outra string) pela string `s2`, como ilustram os exemplos abaixo.

   Importante:

   - Você **pode** assumir que as letras que ocorrem no texto `T` e nas strings `s1` e `s2` são minúsculas.
   - A função `strlen` é a **única** função para manipulação de strings da biblioteca-padrão que pode ser usada nesta questão.

   Exemplos:

   - `T` = "um texto simples.", `s1` = "simples", `s2` = "fácil"
     Texto Final = "um texto <u>fácil</u>."
   - `T` = "esta prova está muito fácil.", `s1` = "muito", `s2` = "absurdamente"
     Texto Final = "esta prova está <u>absurdamente</u> fácil."
   - `T` = "marina come banana.", `s1` = "na", `s2` = "ta"
     Texto Final = "mari<u>ta</u> come ba<u>tata</u>."
   - `T` = "banana", `s1` = "ana", `s2` = "xx"
     Texto Final = "b<u>xx</u>na"
   - `T` = "banana", `s1` = "ana", `s2` = "atata"
     Texto Final = "b<u>atata</u>na"

   Para solucionar o problema acima, resolva os itens (a), (b) e (c) que estão a seguir.

(a) (valor 1.0 ponto)

Abaixo descrevemos o "esqueleto" do programa que troca todas as ocorrências de uma string `s1` na string (texto) `T` por uma string `s2`. Neste item você deve observar o que faz a função `TrocaMaiorMenor` e a função `main` e:

- **Completar** o que falta no cabeçalho da função `TrocaMaiorMenor` (faltam o tipo da função, o nome dos parâmetros e seus respectivos tipos);
- **Completar** o que falta no corpo da função `main` (leitura das strings `s1`, `s2` e `T` e impressão da string `T` após a troca).

As funções `ProcuraString` e `TrocaMenorMaior` que aparecem no esqueleto do programa abaixo serão definidas nos itens (b) e (c) desta questão, respectivamente.

```
#include <string.h>
#include <stdio.h>
#define TextoMax 500
#define StrMax   100

... ProcuraString(...) /* sera feita no item (b) */ { ...}
... TrocaMenorMaior(...) /* sera feita no item (a) */ { ...}
```

/* Complete o cabeçalho da função abaixo */

```
_________TrocaMaiorMenor(______________________________________)
  { int i = 0 , j, ind, existe_string, tam_s1, tam_s2;
    tam_s1 = strlen(s1);  /* tamanho da string s1 */
    tam_s2 = strlen(s2);  /* tamanho da string s2 */
    while (T[i] != '\0')
     { existe_string = ProcuraString(s1, T, i, &ind) ;
       i = ind ;
       if (existe_string == 1)
        { /* troca s1 por s2 no texto */
          j = 0 ;
          while (j < tam_s2)
           { T[ind] = s2[j] ;
             ind++ ; j++; }

          /* atualiza i depois da troca *
          i = ind;

          /* desloca o texto p/ esquerda *
          j = ind + (tam_s1 - tam_s2);
          while (T[j - 1] != '\0')
           { T[ind] = T[j];
             ind++ ; j++; }
        }
     }
  }
```

```
void main(void)
  { char T[TextoMax], s1[StrMax], s2[StrMax];

    /*leitura do texto T e strings s1 e s2*/
```

/*complete aqui com a leitura das strings */

```
    if (strlen(s1) >= strlen(s2)) /*troca de s1 por s2 no texto*/
      TrocaMaiorMenor(s1,s2,T);
    else
      TrocaMenorMaior(s1,s2,T);
    /*imprime texto final*/
    printf("Texto Final:\n");
```

/* complete aqui com a impressão da string T */

```
  }
```

(b) (valor 1.5 ponto)

Faça a função ProcuraString (usada anteriormente pela função TrocaMaiorMenor). A função ProcuraString recebe como parâmetros (veja a chamada da função feita na função ProcuraString):

- uma string s;
- uma string T;
- um inteiro inicio; e
- o endereço endint de uma variável inteira (endint também é dita uma *variável do tipo endereço* ou *apontador*);

e retorna 1 se a string s ocorre na string T, a partir da posição inicio. Caso s não ocorra em T, a partir da posição inicio, a função deve retornar 0.

Além disso, na variável cujo endereço está em endint deverá ser devolvida:

i. a posição de T onde começa a primeira ocorrência da string s, caso s **ocorra** na string T (a partir da posição inicio);

ii. a posição de T onde se encontra o '\0', caso s **não ocorra** em T (a partir da posição inicio).

Exemplo: Suponha que T seja a string a seguir, contendo uma frase de comprimento 42:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
i	r	e	i		b	e	m		n	a		p	r	o	v	a		p	o	i	s		a	
p	r	o	v	a		e	s	t	a		f	a	c	i	l	!	\0							
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

Suponha ainda que s é a string abaixo, contendo uma palavra com 5 caracteres:

0	1	2	3	4	5
p	r	o	v	a	\0

A chamada da função ProcuraString com T e s como descritos acima e inicio $= 0$, retorna 1 e devolve 12 na variável cujo endereço está em endint. Para inicio $= 17$, a função retorna 1 e devolve 25 na variável cujo endereço está em endint. Finalmente, para inicio $= 30$, a função retorna 0 e devolve 42 na variável cujo endereço está em endint.

(c) (valor 1.5 ponto)

Atenção: Mesmo que você **não** tenha feito a função `ProcuraString` do item anterior você **poderá** fazer este item. Para isto basta declarar apenas o protótipo (cabeçalho) da função `ProcuraString`.

Faça a função `TrocaMenorMaior` (usada anteriormente pela função `main`) que, usando **obrigatoriamente** a função `ProcuraString` do item anterior, troca as ocorrências da string `s1` na string `T` pela string `s2`. A função `TrocaMenorMaior` recebe como parâmetros (veja a chamada da função feita na função `main`):

- uma string `s1`;
- uma string `s2`; e
- uma string `T`.

Observe que esta função é chamada na função `main` quando sabemos que o comprimento da string `s2` é maior que o comprimento da string `s1` (ou seja `strlen(s1) < strlen(s2)`).

Importante: Você **pode** supor que o comprimento da string `T` é suficientemente grande para que todas as trocas de `s1` por `s2` sejam feitas sem problema. Ou seja, após as trocas o comprimento da string `T` não terá ultrapassado o limite `TextoMax`.

Exemplo: Suponha que `T` seja a string a seguir, contendo uma frase de comprimento 42:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
i	r	e	i		b	e	m		n	a		p	r	o	v	a		p	o	i	s		a	
p	r	o	v	a		e	s	t	a		f	a	c	i	l	!	\0							
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

Suponha ainda que `s1` é a string abaixo, contendo uma palavra com 5 caracteres:

0	1	2	3	4	5
p	r	o	v	a	\0

e `s2` é a string

0	1	2	3	4	5	6	7
c	o	r	r	i	d	a	\0

A chamada da função `TrocaMenorMaior` com os parâmetros `s1`, `s2` e `T` como descritos acima deve alterar `T` de tal maneira que ao final da função o conteúdo da string `T` seja:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
i	r	e	i		b	e	m		n	a		c	o	r	r	i	d	a		p	o	i	s	
a		c	o	r	r	i	d	a		e	s	t	a		f	a	c	i	l	!	\0			
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

3. (valor 1.5 + 1.0 + 1.0 = 3.5 pontos)
Nesta questão você fará um programa em `C` que leia:

- um inteiro n $(0 < n < 100)$;
- uma matriz de inteiros $A_{n\times n}$;

e que

- calcule a rotação de 90 graus da matriz A no sentido horário (veja o exemplo abaixo);
- preencha com zeros todos os elementos acima da diagonal secundária; e
- imprima a matriz obtida ao final,

como ilustram os exemplos abaixo.

Exemplos:

Matriz A Inicial	**Matriz A Final**
$\begin{pmatrix} 12 & 15 & 3 \\ 10 & 30 & 73 \\ 55 & 12 & 65 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 12 \\ 0 & 30 & 15 \\ 65 & 73 & 3 \end{pmatrix}$

Matriz A Inicial	**Matriz A Final**
$\begin{pmatrix} 0 & 15 & 3 & 13 \\ 40 & 36 & 44 & 20 \\ 28 & 85 & 29 & 66 \\ 77 & 40 & 15 & 18 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 36 & 15 \\ 0 & 29 & 44 & 3 \\ 18 & 66 & 20 & 13 \end{pmatrix}$

Para solucionar o problema acima, resolva os itens (a), (b) e (c) que estão a seguir.

(a) (valor 1.5 ponto)
Faça uma função `Rotacao` que recebe como parâmetros:

- um inteiro n; e
- uma matriz de inteiros $A_{n\times n}$,

e calcule a rotação de 90 graus da matriz A no sentido horário, devolvendo a matriz resultante na própria matriz A.

Exemplo:

Matriz A Inicial	**Matriz A Final**
$\begin{pmatrix} 12 & 15 & 3 \\ 10 & 30 & 73 \\ 55 & 12 & 65 \end{pmatrix}$	$\begin{pmatrix} 55 & 10 & 12 \\ 12 & 30 & 15 \\ 65 & 73 & 3 \end{pmatrix}$

(b) (valor 1.0 ponto)

Faça uma função `ZerosAcimaDiag` que recebe como parâmetros:

- um inteiro n; e
- uma matriz de inteiros $A_{n \times n}$,

e preenche com zeros todas as posições de A acima da diagonal secundária.

Exemplo:

Matriz A Inicial	**Matriz A Final**
$\begin{pmatrix} 55 & 10 & 12 \\ 12 & 30 & 15 \\ 65 & 73 & 3 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 12 \\ 0 & 30 & 15 \\ 65 & 73 & 3 \end{pmatrix}$

(c) (valor 1.0 ponto)

Atenção: Mesmo que você **não** tenha feito as funções `Rotacao` e `ZerosAcimaDiag` dos itens anteriores você **poderá** fazer este item. Para isto basta declarar apenas os protótipos (cabeçalhos) dessas funções.

Faça um programa em `C` que leia:

- um inteiro n ($0 < n < 100$);
- uma matriz de inteiros $A_{n \times n}$;

e que, usando **obrigatoriamente** as funções `Rotacao` e `ZerosAcimaDiag` dos itens anteriores:

- calcule a rotação de 90 graus da matriz A no sentido horário;
- preencha com zeros todos os elementos acima da diagonal secundária; e
- imprima a matriz obtida ao final.

Observação: Não é necessário reescrever as funções dos itens anteriores.