

**MAC2166 – Introdução à Computação para Engenharia**  
ESCOLA POLITÉCNICA  
Primeira Prova – 05 de abril de 2010

Nome: \_\_\_\_\_

Assinatura: \_\_\_\_\_

Nº USP: \_\_\_\_\_ Turma: \_\_\_\_\_

Professor: \_\_\_\_\_

**Instruções:**

1. Não destaque as folhas deste caderno.
2. A prova consta de 4 questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
3. A prova pode ser feita a lápis. Cuidado com a legibilidade.
4. Qualquer questão pode ser resolvida em qualquer página. Se a questão não está na página correspondente ao enunciado basta indicar isto na página e escrever QUESTÃO X em letras ENORMES antes da solução.
5. Não é necessário apagar rascunhos no caderno de questões.
6. Não é permitido o uso de folhas avulsas para rascunho.
7. Não é permitido o uso de calculadoras.
8. Não é permitido a consulta a livros, apontamentos ou colegas.

**DURAÇÃO DA PROVA: 2 horas**

Questão	Valor	Nota
1	2.5	
2	2.5	
3	2.5	
4	2.5	
Total	10.0	

## Questão 1

Simule a execução do programa abaixo, destacando a sua saída. A saída do programa consiste de tudo que resulta dos comandos `printf`. **Para efeito de correção só será considerada a saída do programa.**

```
1 #include <stdio.h>
2
3 int main(){
4     int nusp, d1, d2, a, b, i, j, soma, cont;
5
6     printf ("Entre com o seu no. USP: ");
7     scanf ("%d", &nusp); /* use o seu numero USP como dado de entrada */
8     printf ("nusp = %d\n", nusp);
9
10    d1 = nusp % 10;
11    nusp = nusp/10;
12    d2 = nusp % 10;
13    soma = d1 + d2;
14
15    printf("d1 = %d, d2 = %d, \nsoma inicial = %d", d1, d2, soma);
16
17    if (soma < 10)
18        soma = soma + 10;
19
20    printf("\nsoma parcial = %d\n", soma);
21
22    a = soma % 4 + 2;
23    b = 6 - a;
24    printf("a = %d, b = %d\n", a, b);
25
26    cont = 1;
27    i = 1;
28    j = 1;
29    while (i < a && cont <= 4) {
30        if (j < b) {
31            cont = cont + 1;
32            soma = soma + i*b;
33            printf("i = %d, j = %d, soma = %d\n", i, j, soma);
34            j = j + 1;
35        }
36        else {
37            soma = soma - b;
38            printf("i = %d, j = %d, soma = %d\n", i, j, soma);
39            i = i + 1;
40        }
41    }
42
43    printf("i = %d, j = %d, cont = %d\n", i, j, cont);
44
45    return 0;
46 }
```

Dados para a simulação: **o seu número USP**

main								

Saída do programa

## Questão 2

Pretende-se escrever um programa que calcule o menor elemento par de uma seqüência de inteiros. Para tanto, o programa deve ler um inteiro  $n$ , tal que  $n > 0$ , bem como ler uma seqüência de  $n$  números inteiros que **contém pelo menos um número par**, calculando e imprimindo então o menor número par dentre os elementos da seqüência lida. Por exemplo, para  $n = 5$ , e a seguinte seqüência fornecida como entrada

3 4 2 1 6

o valor a ser impresso com `printf` na saída é 2. Diversas propostas de soluções abaixo são feitas e você deve indicar no quadro correspondente quais são corretas e quais são incorretas. Se você considerar que está incorreto, então você **deve** exibir valores para uma seqüência com  $n = 4$  inteiros que comprovem que o trecho está incorreto. Neste caso, também **deve** exibir o valor *de fato* impresso com `printf` (colocar  $\infty$  se julgar que entra em loop infinito ou tenta ler dados além da conta) e o valor *que deveria ter sido* impresso.

<pre>scanf("%d",&amp;menorpar); for(i=1; i&lt;n; i++){   scanf("%d",&amp;x);   if ( x&lt;menorpar )     menorpar=x; } printf("%d\n",menorpar);</pre>	<input type="checkbox"/> Funciona sempre. <input type="checkbox"/> Falha para <input type="text"/> , <input type="text"/> , <input type="text"/> , <input type="text"/> , pois imprime <input type="text"/> , mas deveria imprimir <input type="text"/> .
<pre>menorpar=99; for( i=0; i&lt;n; i++){   scanf("%d",&amp;x);   if ( (x%2==0) &amp;&amp; (x&lt;menorpar) )     menorpar=x; } printf("%d\n",menorpar);</pre>	<input type="checkbox"/> Funciona sempre. <input type="checkbox"/> Falha para <input type="text"/> , <input type="text"/> , <input type="text"/> , <input type="text"/> , pois imprime <input type="text"/> , mas deveria imprimir <input type="text"/> .
<pre>scanf("%d",&amp;menorpar); for(i=1; i&lt;n; i++){   scanf("%d",&amp;x);   if ( (x%2==0) &amp;&amp; (x&lt;menorpar) )     menorpar=x; } printf("%d\n",menorpar);</pre>	<input type="checkbox"/> Funciona sempre. <input type="checkbox"/> Falha para <input type="text"/> , <input type="text"/> , <input type="text"/> , <input type="text"/> , pois imprime <input type="text"/> , mas deveria imprimir <input type="text"/> .
<pre>i=0; scanf("%d",&amp;menorpar); while(menorpar%2 != 0)   scanf("%d",&amp;menorpar); while(i&lt;n){   scanf("%d",&amp;x);   if ( (x%2==0) &amp;&amp; (x&lt;menorpar) )     menorpar=x; } printf("%d\n",menorpar);</pre>	<input type="checkbox"/> Funciona sempre. <input type="checkbox"/> Falha para <input type="text"/> , <input type="text"/> , <input type="text"/> , <input type="text"/> , pois imprime <input type="text"/> , mas deveria imprimir <input type="text"/> .
<pre>menorpar= 1; i=0; while(menorpar%2 != 0){   scanf("%d",&amp;menorpar);   i++; } while(i&lt;n){   scanf("%d",&amp;x);   if ( (x%2==0) &amp;&amp; (x&lt;menorpar) )     menorpar=x;   i++; } printf("%d\n",menorpar);</pre>	<input type="checkbox"/> Funciona sempre. <input type="checkbox"/> Falha para <input type="text"/> , <input type="text"/> , <input type="text"/> , <input type="text"/> , pois imprime <input type="text"/> , mas deveria imprimir <input type="text"/> .

### Questão 3

Um número natural é dito perfeito se for igual à soma de todos os seus divisores positivos próprios (exclui-se ele mesmo). Por exemplo, o número 6 é um número perfeito, pois:

$$6 = 1 + 2 + 3.$$

O próximo número perfeito é

$$28 = 1 + 2 + 4 + 7 + 14.$$

Escreva um programa em C que lê um inteiro  $n$  positivo e imprime todos os números perfeitos menores ou iguais a  $n$ .

## Questão 4

No EP1, V. fez um programa para que dada uma seqüência de números com 9 dígitos (1 a 9), verificar se cada um deles forma um quadrado mágico. Usando idéias parecidas com o que V. já implementou no EP1, faça um programa para que dado um número inteiro  $n$  ( $n > 0$ ), representando a quantidade de elementos de uma seqüência, e  $n$  números da seqüência, cada um com 9 dígitos (1 a 9) representando um quadrado, verificar se a soma dos elementos das extremidades das diagonais principal e secundária são iguais ao elemento central do quadrado para cada número da seqüência. Por exemplo: o quadrado

2	7	4
9	8	1
4	3	6

Possui as somas das extremidades das diagonais iguais ao elemento central, porque  $2 + 6 = 8$  e  $4 + 4 = 8$ . Por outro lado, o quadrado

3	7	4
9	8	1
4	3	6

Não possui as somas das extremidades das diagonais iguais ao elemento central, porque  $3 + 6 \neq 8$ .

Como exemplo de uso do seu programa, poderíamos ter:  $n = 2$  e os quadrados dos exemplos acima. Neste caso, o seu programa deverá ter como entrada e saída:

Digite a quantidade de elementos da sequencia: 2

Entre com um quadrado: 274981436

O quadrado possui as somas das extremidades das diagonais iguais ao elemento central

Entre com um quadrado: 374981436

O quadrado NAO possui as somas das extremidades das diagonais iguais ao elemento central

**Note que não é para verificar se é um quadrado mágico, apenas as somas dos elementos das extremidades das diagonais.**

```
# include <stdio.h>

int main(){
    int n,          /* quantidade de elementos da sequencia */
        num,       /* numero lido */

                                ; /* outras variaveis*/

    /* leitura da quantidade de elementos da sequencia */
    printf ("\n Digite a quantidade de elementos da sequencia: ");
    scanf ("%d", &n);

    return 0;
}
```