

MAC2166 – Introdução à Computação para Engenharia
ESCOLA POLITÉCNICA
Primeira Prova – 04 de abril de 2011

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	2.0	
2	2.0	
3	3.0	
4	3.0	
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, a1, a2, b, c, d, i, j, total, 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    nusp = nusp % 100;
11    a1 = nusp % 3;
12    nusp = nusp / 3;
13    a2 = nusp % 3;
14    total = a1 + a2 + 10;
15
16    printf ("nusp = %d, a1 = %d, a2 = %d, total = %d\n", nusp, a1, a2, total);
17
18    b = total / 8;
19    c = total - 10;
20    d = total % 2;
21
22    printf ("b = %d, c = %d, d = %d\n", b, c, d);
23
24    cont = 0;
25    i = b;
26    j = c + 5;
27    while (i <= j && cont < b + 3) {
28        c = total % 5;
29        if ( c % 2 != 0 || c == 2) {
30            printf ("A\n");
31            printf ("i = %d, j = %d, c = %d\n", i, j, c);
32            i = i + 1;
33        }
34        else {
35            printf ("B\n");
36            printf ("i = %d, j = %d, c = %d\n", i, j, c);
37            j = j - 1;
38        }
39    }
40    cont = cont + 1;
41    total = total / 2;
42    printf ("cont = %d, total = %d\n", cont, total);
43 }
44
45 printf ("i = %d, j = %d, c = %d, cont = %d, total = %d\n", i, j, c, cont, total);
46
47 return 0;
48 }
```

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

main									

Saída do programa

Questão 2

No EP1, você fez um programa para verificação do jogo SOS. Neste EP, um número inteiro n entre 0 e 19682 era utilizado para representar um tabuleiro da seguinte forma:

- (a) Primeiramente, encontrava-se para n a sua representação na base 3. Por exemplo, para $n = 4341$, os dígitos de sua representação na base 3 (marcados em negrito) são

$$n = 4341 = \mathbf{0} \times 3^0 + \mathbf{1} \times 3^1 + \mathbf{2} \times 3^2 + \mathbf{1} \times 3^3 + \mathbf{2} \times 3^4 + \mathbf{2} \times 3^5 + \mathbf{2} \times 3^6 + \mathbf{1} \times 3^7 + \mathbf{0} \times 3^8$$

- (b) Depois, usamos a seguinte codificação para numeração das posições do tabuleiro,

0	1	2
3	4	5
6	7	8

onde cada número marcado nesta tabela corresponde a um expoente de uma potência de 3. Da mesma forma como você fez no EP1, o correspondente **coeficiente** da potência de 3 que compõe o inteiro n deve ser colocado neste tabuleiro. Por exemplo, o número $n = 4341$ corresponde ao tabuleiro

0	1	2
1	2	2
2	1	0

Nesta questão, estamos interessados em encontrar somente o número que aparece na linha do meio do tabuleiro. No exemplo anterior, o número que aparece no meio do tabuleiro é o inteiro 122. Outros exemplos:

para $n = 17458$, temos o tabuleiro <table style="border-collapse: collapse; margin: auto;"> <tr> <td style="border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px;">1</td> <td style="border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px;">2</td> <td style="border-bottom: 1px solid black; padding: 5px;">1</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">1</td> <td style="border-right: 1px solid black; padding: 5px;">2</td> <td style="padding: 5px;">2</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">2</td> <td style="border-right: 1px solid black; padding: 5px;">1</td> <td style="padding: 5px;">2</td> </tr> </table> com o inteiro 122 na linha do meio	1	2	1	1	2	2	2	1	2	para $n = 4342$, temos o tabuleiro <table style="border-collapse: collapse; margin: auto;"> <tr> <td style="border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px;">1</td> <td style="border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px;">1</td> <td style="border-bottom: 1px solid black; padding: 5px;">2</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">1</td> <td style="border-right: 1px solid black; padding: 5px;">2</td> <td style="padding: 5px;">2</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">2</td> <td style="border-right: 1px solid black; padding: 5px;">1</td> <td style="padding: 5px;">0</td> </tr> </table> com o inteiro 122 na linha do meio	1	1	2	1	2	2	2	1	0	para $n = 12056$, temos o tabuleiro <table style="border-collapse: collapse; margin: auto;"> <tr> <td style="border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px;">2</td> <td style="border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px;">1</td> <td style="border-bottom: 1px solid black; padding: 5px;">1</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">2</td> <td style="border-right: 1px solid black; padding: 5px;">1</td> <td style="padding: 5px;">1</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">1</td> <td style="border-right: 1px solid black; padding: 5px;">2</td> <td style="padding: 5px;">1</td> </tr> </table> com o inteiro 211 na linha do meio	2	1	1	2	1	1	1	2	1	para $n = 17143$, temos o tabuleiro <table style="border-collapse: collapse; margin: auto;"> <tr> <td style="border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px;">1</td> <td style="border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px;">2</td> <td style="border-bottom: 1px solid black; padding: 5px;">2</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">1</td> <td style="border-right: 1px solid black; padding: 5px;">1</td> <td style="padding: 5px;">1</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">2</td> <td style="border-right: 1px solid black; padding: 5px;">1</td> <td style="padding: 5px;">2</td> </tr> </table> com o inteiro 111 na linha do meio	1	2	2	1	1	1	2	1	2
1	2	1																																					
1	2	2																																					
2	1	2																																					
1	1	2																																					
1	2	2																																					
2	1	0																																					
2	1	1																																					
2	1	1																																					
1	2	1																																					
1	2	2																																					
1	1	1																																					
2	1	2																																					

Pretende-se escrever um programa que, dado um inteiro entre 0 e 19682, extraia o número inteiro da linha do meio. Diversas propostas de solução são feitas abaixo e você deve indicar no quadro correspondente quais são corretas e quais são incorretas. Se você considerar que está incorreta, então você **deve** exibir um inteiro n que comprove que o trecho está incorreto. Neste caso, também **deve** exibir o valor *de fato* impresso pelo programa fornecido com `printf` (colocar ∞ se julgar que o programa entra em loop infinito ou tenta ler dados além da conta) e o valor *que deveria ter sido* impresso se o programa estivesse correto. Considere que nos trechos de programa abaixo todas as variáveis utilizadas são do tipo `int` e nenhum dos trechos possui erro de sintaxe, ou seja, todos os trechos compilam sem erro.

<pre>scanf("%d",&n); cont = 0; while (n>0 && cont<3) {n = n / 3; cont++;} linhadomeio = 0; cont = 0; while (n>0 && cont<3) { d = n % 3; linhadomeio = linhadomeio * 10 + d; n = n / 3; cont++; } printf("%d\n",linhadomeio);</pre>	<input type="checkbox"/> Funciona sempre. <input type="checkbox"/> Falha para $n = \boxed{}$, pois imprime $\boxed{}$, mas deveria imprimir $\boxed{}$.
<pre>scanf("%d",&n); for (i=0; i<3; i++) n = n / 3; d5 = n % 3; d4 = (n / 3) % 3; d3 = (n / 3) % 3; linhadomeio = d3*100+d4*10+d3; printf("%d\n",linhadomeio);</pre>	<input type="checkbox"/> Funciona sempre. <input type="checkbox"/> Falha para $n = \boxed{}$, pois imprime $\boxed{}$, mas deveria imprimir $\boxed{}$.
<pre>scanf("%d",&n); for (i=0; i<3 n>0; i++) n = n / 3; linhadomeio = 0; for (i=0; i<3 n>0; i++) { d = n - 3 * (n / 3); linhadomeio = linhadomeio * 10 + d; n = n / 3; } printf("%d\n",linhadomeio);</pre>	<input type="checkbox"/> Funciona sempre. <input type="checkbox"/> Falha para $n = \boxed{}$, pois imprime $\boxed{}$, mas deveria imprimir $\boxed{}$.
<pre>scanf("%d",&n); n = n / 27; linhadomeio = 0; for (i=0; i<3 && n>0; i++) { d = n % 3; linhadomeio = linhadomeio * 10 + d; n = n / 3; } printf("%d\n",linhadomeio);</pre>	<input type="checkbox"/> Funciona sempre. <input type="checkbox"/> Falha para $n = \boxed{}$, pois imprime $\boxed{}$, mas deveria imprimir $\boxed{}$.
<pre>scanf("%d",&n); n = n / 27; d5 = n % 3; d4 = (n / 3) % 3; d3 = (n / 9) % 3; linhadomeio = d5*100+d4*10+d3; printf("%d\n",linhadomeio);</pre>	<input type="checkbox"/> Funciona sempre. <input type="checkbox"/> Falha para $n = \boxed{}$, pois imprime $\boxed{}$, mas deveria imprimir $\boxed{}$.

Questão 3

Dado um número n , $n > 0$, verificar se esse número é um quadrado perfeito ou não. Caso seja, imprimir de qual número ele é um quadrado.

Exemplo 1:

Dado 81, seu programa deve imprimir:
81 e o quadrado perfeito de 9.

Exemplo 2:

Dado 19, seu programa deve imprimir:
19 não é um quadrado perfeito.

Questão 4

Casal perfeito: Dadas as datas de aniversário de um homem e de uma mulher, diz-se que o casal é perfeito se o número mágico dos dois é igual. O número mágico é obtido por repetidamente somar-se os dígitos até se obter um número com um único dígito. Seu programa deve receber duas datas de aniversário, cada uma dada por um número inteiro positivo no formato ddmmaaaa...a e imprimir se o casal é perfeito ou não. Note que seu programa deve aceitar também, datas com anos futurísticos como o ano de 10177, por exemplo (e não somente anos com 4 dígitos).

Exemplo:

Data de aniversário do homem=19011981

$1+9+0+1+1+9+8+1=30$

$3+0=3$

Número mágico do homem=3

Data de aniversário da mulher=20021979

$2+0+0+2+1+9+7+9=30$

$3+0=3$

Número mágico da mulher=3

Logo, esse casal é um casal perfeito.