ACH2001-Introdução à Ciência da Computação 1 – Prova 1 -28.04.2011-Duração 1h45min.

Nome: no. USP

Há várias provas diferentes. Devolva esta folha, preenchida, junto com a resolução. Caso seja detectada cola, a nota será zero.

1- Valor total: 2pt -) **Cite** três paradigmas de linguagens de programação(0,1 pt cada). **Explique** sucintamente(0,1 pt cada) e **cite** ao menos uma característica importante e diferente das citadas a seguir (0,1 pt cada) de cada paradigma. No paradigma orientado a objetos, **explique** ocultamento, encapsulamento e sua relação com especificadores de acesso (1,1pt).

2- Valor total: 2pt -) **Escreva** um programa HIPO que armazene em cada variável um dígito e conte de 77 a 00 (decrescente) em octal.

obs.: O HIPO teve duas instruções substituídas, em negrito na tabela abaixo.

Código da operação	Mnemônico	Explicação
11	CEA	Copie o conteúdo do endereço EE no acumulador. (AC recebe [EE]).
12	CAE	Copie o conteúdo do acumulador no endereço EE. (EE recebe [AC])
21	SOM	Some o conteúdo do endereço EE com o conteúdo do acumulador e guarde o resultado no acumulador. (AC recebe [AC] + [EE])
22	SUB	Subtraia o conteúdo do endereço EE do conteúdo do acumulador e guarde o resultado no acumulador. (AC recebe [AC] - [EE])
23	MUL	Multiplique o conteúdo do endereço EE com o conteúdo do acumulador e guarde o resultado no acumulador. (AC recebe [AC] * [EE])
24	DIV	Divide o conteúdo do acumulador pelo conteúdo do endereço EE e guarde o resultado no acumulador. (AC recebe [AC] / [EE])
25	MOD	[AC] recebe o resto da divisão [AC] / [EE].
31	LER	Leia um número e guarde-o no endereço EE. (EE recebe o valor lido)
41	IMP	Imprima o conteúdo do endereço EE.
50	NOP	Nenhuma operação é efetuada.
51	DES	Desvie a execução para o endereço EE, i.e. AI recebe EE.
52	DPO	Se o conteúdo do acumulador for maior do que zero, desvie a execução para o endereço EE. (Se [AC] > 0, AI recebe EE).
53	DPZ	Se o conteúdo do acumulador for maior ou igual a zero, desvie a execução para o endereço EE. (Se [AC] >= 0, AI recebe EE).
54	DNE	Se o conteúdo do acumulador for menor do que zero, desvie a execução para o endereço EE. (Se [AC] < 0, AI recebe EE.)
55	DNZ	Se o conteúdo do acumulador for menor ou igual a zero, desvie a execução para o endereço EE. (Se [AC] <= 0, AI recebe EE).
56	DDZ	Se o conteúdo do acumulador for diferente de zero, desvie a execução para o endereço EE. (Se [AC] != 0, AI recebe EE).
57	DZZ	Se o conteúdo do acumulador for igual a zero, desvie a execução para o endereço EE. (Se [AC] = 0, AI recebe EE).
58	**IDI**	**Imprima o caracter ('0'..'9','A'..'F') correspondente ao conteúdo do acumulador (+0000..+0015). Caso o conteúdo esteja fora da faixa, não imprime nada.**
59	**PUL**	**"Pula uma linha".**
61	ADE	Desloque os dígitos do acumulador uma posição à esquerda, desprezando o digito mais significativo.
62	ADD	Desloque os dígitos do acumulador uma posição à direita, desprezando o digito menos significativo.
70	PAR	Pare a execução do programa. OBS. Esta instrução deve ser executada para encerrar a execução do programa.

Q3 – valor total: 1pt -) **Apresente** os números $(75)_8$, $(51)_{10}$ nas bases 2, 8, 10, 16 (0,1pt cada número em cada base diferente da apresentada no enunciado). **Apresente também** os mesmos números com sinal invertido (negativos) em binário (**6 bits**) e representação complemento de dois (0,2pt cada número).

Q4- valor total:2,5pt -) **Escreva** os métodos calculaTroco (...) (1,5pt) e imprimeNotas () (0,5pt), conforme especificado. **Apresente** o que o programa imprime na tela quando é executado (0,5pt).

```
class Caixa {

    /** Valores de face de cada nota */
    public static final int FaceA=65536, FaceB=4096, FaceC=256, FaceD=16, FaceE=1;

    /** Quantidades de notas */
    public static int A, B, C, D, E;

    /** Imprime na tela a quantidade de notas de cada valor que
      * devem ser devolvidas.
      */
    public void imprimeNotas () {
    }

    /** Recebe o Valor da mercadoria que o cliente comprou e a quantidade
      * de notas de cada valor de face que o cliente entregou. Calcula
      * o troco e armazena em A, B, C, D e E, a quantidade mínima de notas
      * que tem que ser devolvidas ao cliente.
      */

    public void calculaTroco  (int Valor, int NotasA, int NotasB, int NotasC,
                               int NotasD, int NotasE) {
    }
    public static void main (String args[]) {
        Caixa Cx1;

        Cx1=new Caixa();

        Cx1.calculaTroco (24570, 1, 0, 0, 0, 0);
        Cx1.imprimeNotas ();
    }
}
```

Q5 – valor total: 2,5pt -) Em um novo código-fonte, o método main acima foi substituído pelo abaixo, cuja intenção é trabalhar com 3 caixas distintos e independentes (Não há erros de digitação no código abaixo).

1	`public static void main (String args[]) {`
2	`Caixa Cx1, Cx2, Cx3;`
3	
4	`Cx1=new Caixa();`
5	`Cx2=new Caixa();`
6	
7	`Cx3=Cx1;`
8	`Cx3.calculaTroco(24570, 1, 0, 0, 0, 0);`
9	`Cx2.imprimeNotas();`
10	`}`

a-) **Escreva** sucintamente(Max. 1 linha), o que cada linha (2..9) do método main, acima, faz. (0,5pt)

b-) O novo código-fonte compila? Proponha uma correção caso necessário (0,5pt)

c-) O que o programa **imprime** quando é executado? (0,5pt)

d-) Há três caixas distintos e independentes? Explique o motivo. (1pt)