

MAC2166 – Introdução à Computação para Engenharia
ESCOLA POLITÉCNICA
Primeira Prova – 09 de abril de 2012

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 equipamentos eletrônicos.
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 (vale 2,5 pontos)

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.**

```
#include <stdio.h>

int main()
{
    int nusp, a, b, c, d;

    printf ("Entre com o seu no. USP: ");
    scanf ("%d", &nusp); /* use o seu numero USP como dado de entrada */
    printf ("nusp = %d\n", nusp);

    a = nusp % 100;
    b = nusp % 10;
    c = nusp % 1;
    d = 11 - b;
    printf("a = %d b = %d c = %d d = %d\n", a, b, c, d);

    while (b >= 0 || d >= 0) {
        if (b > d) {
            b = b - 6;
            printf("Verdadeiro b = %d d = %d\n", b, d);
        }
        else {
            d = d - 6;
            printf("Falso b = %d d = %d\n", b, d);
        }
    }

    return 0;
}
```

Você pode usar a tabela abaixo como bem entender. Cada turma está habituada a simular de maneira diferente, fazendo tabelas com “caras” diferentes da abaixo.

main									

A solução depende do seu número USP.
A seguir está a solução para o número USP 1234567.

Saída do programa
Entre com o seu no. USP: 1234567 nusp = 1234567 a = 67 b = 7 c = 0 d = 4 Verdadeiro b = 1 d = 4 Falso b = 1 d = -2 Verdadeiro b = -5 d = -2

Questão 2 (vale 2,5 pontos)

Um número natural na base decimal pode ser escrito na base binária seguindo o exemplo apresentado a seguir:

$$23 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

Logo, 23 em binário é 10111.

Pretende-se escrever um programa em C que lê um inteiro $n \geq 0$ na base decimal e imprime a representação de n na base binária. Diversas propostas de solução são feitas a seguir 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 $n \geq 0$ 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); aux = n; bin = 0; for (pot=1;aux>0;pot=pot*10) { dig = aux % 2; aux = aux / 2; bin = bin + dig * pot; } printf("%d\n", bin);</pre>	<p><input checked="" type="checkbox"/> Funciona sempre.</p> <p><input type="checkbox"/> Falha para $n =$ <input type="text"/>, pois imprime <input type="text"/>, mas deveria imprimir <input type="text"/>.</p>
<pre>scanf("%d", &n); aux = n; for (pot=1;aux>0;pot=pot*10) { dig = aux % 2; aux = aux / 2; printf("%d",dig); } if (n==0) printf("0"); printf("\n");</pre>	<p><input type="checkbox"/> Funciona sempre.</p> <p><input checked="" type="checkbox"/> Falha para $n =$ <input type="text" value="2"/>, pois imprime <input type="text" value="01"/>, mas deveria imprimir <input type="text" value="10"/>.</p>
<pre>scanf("%d", &n); aux = n; bin = 0; pot = 1; while (aux>=0) { dig = aux % 2; aux = aux / 2; bin = bin + dig * pot; pot=pot*10; } printf("%d\n", bin);</pre>	<p><input type="checkbox"/> Funciona sempre.</p> <p><input checked="" type="checkbox"/> Falha para $n =$ <input type="text" value="0,1,2,..."/>, pois imprime <input type="text" value="∞"/>, mas deveria imprimir <input type="text" value="0,1,10,..."/>.</p>
<pre>scanf("%d", &n); bin = 0; dig = 1; for (pot=1;dig>0;pot=pot*10) { dig = n % 2; n = n / 2; bin = bin + dig * pot; } printf("%d\n", bin);</pre>	<p><input type="checkbox"/> Funciona sempre.</p> <p><input checked="" type="checkbox"/> Falha para $n =$ <input type="text" value="2"/>, pois imprime <input type="text" value="0"/>, mas deveria imprimir <input type="text" value="10"/>.</p>

Questão 3 (vale 2,5 pontos)

Um caixa de banco tem somente notas de a e b reais para fornecer aos seus clientes. Quando um cliente chega com um cheque no valor de c reais, o caixa precisa fazer umas contas para decidir se é possível descontar o cheque ou não.

Escreva um programa em C que resolva o problema do caixa de banco: o programa recebe como entrada os inteiros positivos a , b e c e devolve como saída **TODAS** as combinações possíveis de notas de a reais e notas de b reais, totalizando c reais, ou uma mensagem informando que não é possível descontar o cheque.

Exemplos:

– Para $a=5$, $b=3$ e $c=11$ seu programa deve imprimir:

1 nota de 5 e 2 notas de 3

– Para $a=3$, $b=5$ e $c=30$ seu programa deve imprimir:

10 notas de 3 e 0 notas de 5

5 notas de 3 e 3 notas de 5

0 notas de 3 e 6 notas de 5

– Para $a=5$, $b=3$ e $c=7$ seu programa deve imprimir:

Nao eh possivel descontar o cheque

– Para $a=4$, $b=6$ e $c=121$ seu programa deve imprimir:

Nao eh possivel descontar o cheque

– Para $a=4$, $b=7$ e $c=17$ seu programa deve imprimir:

Nao eh possivel descontar o cheque

```

/*
 * Solucao 1: curta e grossa
 *          usa dois lacos 'for' encaixados
 *
 */

#include <stdio.h>

#define SIM 1
#define NAO 0

int main()
{
    int a, b;      /* valores das notas */
    int c;        /* valor do cheque  */
    int na;       /* numero de notas de a reais */
    int nb;       /* numero de notas de b reais */
    int possivel; /* indica e foi possivel descontar o cheque */

    printf("Digite os valores das notas: ");
    scanf("%d %d", &a, &b);

    printf("Digite o valor do cheque: ");
    scanf("%d", &c);

    possivel = NAO;
    for (na = 0; na <= c/a; na++)
    {
        for (nb = 0; nb <= c/b; nb++)
        {
            if (na*a + nb*b == c)
            {
                printf("%d nota(s) de %d e %d nota(s) de %d\n",na,a,nb,b);
                possivel = SIM;
            }
        }
    }

    if (possivel == NAO)
    {
        printf("Nao eh possivel descontar o cheque\n");
    }

    return 0;
}

```

```

/*
 * Solucao 2: identica a solucao 1 sem algumas chaves superfluas
 */

#include <stdio.h>

#define SIM 1
#define NAO 0

int main()
{
    int a, b;      /* valores das notas */
    int c;        /* valor do cheque */
    int na;       /* numero de notas de a reais */
    int nb;       /* numero de notas de b reais */
    int possivel; /* indica e foi possivel descontar o cheque */

    printf("Digite os valores das notas: ");
    scanf("%d %d", &a, &b);

    printf("Digite o valor do cheque: ");
    scanf("%d", &c);

    possivel = NAO;
    for (na = 0; na <= c/a; na++)
        for (nb = 0; nb <= c/b; nb++)
            if (na*a + nb*b == c)
                {
                    printf("%d nota(s) de %d e %d nota(s) de %d\n",na,a,nb,b);
                    possivel = SIM;
                }

    if (possivel == NAO) printf("Nao eh possivel descontar o cheque\n");

    return 0;
}

```

```

/*
 * Solucao 3: identica a solucao 1 com 'while' no lugar de 'for'
 */
#include <stdio.h>

#define SIM 1
#define NAO 0

int main()
{
    int a, b;    /* valores das notas */
    int c;      /* valor do cheque  */
    int na;     /* numero de notas de a reais */
    int nb;     /* numero de notas de b reais */
    int possivel; /* indica e foi possivel descontar o cheque */

    printf("Digite os valores das notas: ");
    scanf("%d %d", &a, &b);

    printf("Digite o valor do cheque: ");
    scanf("%d", &c);

    possivel = NAO;
    na = 0;
    while (na <= c/a)
    {
        nb = 0;
        while (nb <= c/b)
        {
            if (na*a + nb*b == c)
            {
                printf("%d nota(s) de %d e %d nota(s) de %d\n",na,a,nb,b);
                possivel = SIM;
            }
            nb = nb + 1;
        }
        na = na + 1;
    }

    if (possivel == NAO)
    {
        printf("Nao eh possivel descontar o cheque\n");
    }

    return 0;
}

```



```

/*
 * Solucao 4: usa apenas um laço
 */
#include <stdio.h>

#define SIM 1
#define NAO 0

int main()
{
    int a, b;    /* valores das notas */
    int c;      /* valor do cheque  */
    int na;     /* numero de notas de a reais */
    int possivel; /* indica e foi possivel descontar o cheque */

    printf("Digite os valores das notas: ");
    scanf("%d %d", &a, &b);

    printf("Digite o valor do cheque: ");
    scanf("%d", &c);

    possivel = NAO;
    na = 0;
    while (c >= 0)
    {
        if (c%b == 0)
        {
            printf("%d nota(s) de %d e %d nota(s) de %d\n",na,a,c/b,b);
            possivel = SIM;
        }
        c = c - a;
        na = na + 1;
    }

    if (possivel == NAO)
    {
        printf("Nao eh possivel descontar o cheque\n");
    }

    return 0;
}

```

Questão 4 (vale 2,5 pontos)

Escreva um programa em C que lê as configurações dos pinos A, B e C de um quebra-cabeça da Torre de Hanói, e imprime, para cada par de pinos, uma mensagem indicando se o movimento de um pino para outro é válido ou inválido. Considere que o quebra-cabeça **SEMPRE** tem 4 discos.

Exemplo do comportamento do programa:

Digite a configuracao do pino A: 1 0 0 0

Digite a configuracao do pino B: 4 0 0 0

Digite a configuracao do pino C: 3 2 0 0

Saida do programa:

do pino A para o pino B: valido

do pino B para o pino A: invalido

do pino A para o pino C: valido

do pino C para o pino A: invalido

do pino B para o pino C: invalido

do pino C para o pino B: valido

```

/*
 * Solucao 1: topoX == 0 significa que a pilha X esta vazia
 */

#include <stdio.h>

int main () {
    /*
     * Ha tres pinos: o pino A, o pino B e o pino C.
     *
     * As variaveis a1, a2, a3 e a4 informam quais discos estao no pino A.
     * A variavel a1 e a base do pino e a posicao a4 e o topo do pino.
     *
     * As variaveis b1, b2, b3 e b4 tem o mesmo papel mas para o pino B.
     * As variaveis c1, c2, c3 e c4 tem o mesmo papel mas para o pino C.
     *
     * O valor de cada variavel pode ser 0, 1, 2, 3 ou 4, onde
     * 0 significa que nao ha disco e entre 1 e 4 significa que ha um
     * disco com o tamanho equivalente a cada valor.
     */

    int a1, a2, a3, a4; /* guarda a configuracao do pino A */
    int b1, b2, b3, b4; /* guarda a configuracao do pino B */
    int c1, c2, c3, c4; /* guarda a configuracao do pino C */

    int topoA; /* guarda o (valor do) disco no topo do pino A */
    int topoB; /* guarda o (valor do) disco no topo do pino B */
    int topoC; /* guarda o (valor do) disco no topo do pino C */

    /*-----*/
    /* 1. LEIA as configuracoes dos pinos */
    printf("Digite a configuracao do pino A: ");
    scanf("%d %d %d %d",&a1,&a2,&a3,&a4);
    printf("Digite a configuracao do pino B: ");
    scanf("%d %d %d %d",&b1,&b2,&b3,&b4);
    printf("Digite a configuracao do pino C: ");
    scanf("%d %d %d %d",&c1,&c2,&c3,&c4);

    /*-----*/
    /* 2. DETERMINE os discos nos topos dos pinos */

    /*- - - - -*/
    /* 2.1 DETERMINE o disco no topo do pino A */
    topoA = 0; /* pino A esta vazio ate que se prove o contrario */
    if (a4!=0) topoA = a4;
    else if (a3!=0) topoA = a3;
    else if (a2!=0) topoA = a2;
    else if (a1!=0) topoA = a1;

```



```

/*
 * Solucao 2: topoX == 5 significa que a pilha do pino X esta vazia
 */

#include <stdio.h>

int main () {
    /*
     * Ha tres pinos. o pino A, o pino B e o pino C.
     *
     * As variaveis a1, a2, a3 e a4 informam quais discos estao no pino A.
     * A variavel a1 e a base do pino e a posicao a4 e o topo do pino.
     *
     * As variaveis b1, b2, b3 e b4 tem o mesmo papel mas para o pino B.
     * As variaveis c1, c2, c3 e c4 tem o mesmo papel mas para o pino C.
     *
     * O valor de cada variavel pode ser 0, 1, 2, 3 ou 4, onde
     * 0 significa que nao ha disco e entre 1 e 4 significa que ha um
     * disco com o tamanho equivalente a cada valor.
     */

    int a1, a2, a3, a4; /* guarda a configuracao do pino A */
    int b1, b2, b3, b4; /* guarda a configuracao do pino B */
    int c1, c2, c3, c4; /* guarda a configuracao do pino C */

    int topoA; /* guarda o (valor do) disco no topo do pino A */
    int topoB; /* guarda o (valor do) disco no topo do pino B */
    int topoC; /* guarda o (valor do) disco no topo do pino C */

    /*-----*/
    /* 1. LEIA das configuracoes dos pinos */
    printf("Digite a configuracao do pino A: ");
    scanf("%d %d %d %d",&a1,&a2,&a3,&a4);
    printf("Digite a configuracao do pino B: ");
    scanf("%d %d %d %d",&b1,&b2,&b3,&b4);
    printf("Digite a configuracao do pino C: ");
    scanf("%d %d %d %d",&c1,&c2,&c3,&c4);

    /*-----*/
    /* 2. DETERMINE os discos nos topos dos pinos */

    /*- - - - -*/
    /* 2.1 DETERMINE o disco no topo do pino A */
    topoA = 5; /* pino A esta vazio ate que se prove o contrario */
    if (a4!=0) topoA = a4;
    else if (a3!=0) topoA = a3;
    else if (a2!=0) topoA = a2;
    else if (a1!=0) topoA = a1;

```

