

MAC 2166 – Introdução à Computação para Engenharia
ESCOLA POLITÉCNICA – PRIMEIRO SEMESTRE DE 2001
Primeira Prova – 18 de abril de 2001
GABARITO

1. (valor 2,5 pontos)

Simule a execução do programa abaixo, destacando a saída do programa. Considere que a saída do programa consiste de tudo que resulta dos comandos *printf*.

```
#include <stdio.h>
int main()
{
    int a, b, c, m, n, NUSP;

    printf("Digite seu NUSP: ");
    scanf("%d", &NUSP); /* Use o seu n USP */
    n = NUSP % 10;
    printf("Ultimo digito = %d\n", n);
    m = 3*n + 1;
    printf("m = %d\n",m);
    a = m % (n+1);
    printf("a = %d\n",a);
    b = m/(n+1);
    printf("b = %d\n",b);
    c = b*(n+1) + a;
    printf("c = %d\n",c);
    if (c > m)
        printf("Maior\n");
    else
        if (c < m)
            printf("Menor\n");
        else
            if (m > a && m >= b)
                printf("Igual\n");
            else
                printf("Diferente\n");

    while (m > 2 || m % 3 == 1){
        m = m - 6;
        printf("m = %d\n",m);
    }
    return 0;
}
```

SOLUÇÃO: Apenas o último dígito é relevante

Digite seu NUSP: 0

Ultimo digito = 0

$m = 1$

$a = 0$

$b = 1$

$c = 1$

Igual

$m = -5$

Digite seu NUSP: 1

Ultimo digito = 1

$m = 4$

$a = 0$

$b = 2$

$c = 4$

Igual

$m = -2$

Digite seu NUSP: 2

Ultimo digito = 2

$m = 7$

$a = 1$

$b = 2$

$c = 7$

Igual

$m = 1$

$m = -5$

Digite seu NUSP: 3

Ultimo digito = 3

$m = 10$

$a = 2$

$b = 2$

$c = 10$

Igual

$m = 4$

$m = -2$

Digite seu NUSP: 4

Ultimo digito = 4

$m = 13$

$a = 3$

$b = 2$

$c = 13$

Igual

$m = 7$

$m = 1$

$m = -5$

Digite seu NUSP: 5

Ultimo digito = 5

$m = 16$

$a = 4$

$b = 2$

$c = 16$

Igual

$m = 10$

$m = 4$

$m = -2$

Digite seu NUSP: 6

Ultimo digito = 6

$m = 19$

$a = 5$

$b = 2$

$c = 19$

Igual

$m = 13$

$m = 7$

$m = 1$

$m = -5$

Digite seu NUSP: 7

Ultimo digito = 7

$m = 22$

$a = 6$

$b = 2$

$c = 22$

Igual

$m = 16$

$m = 10$

$m = 4$

$m = -2$

Digite seu NUSP: 8

Ultimo digito = 8

$m = 25$

$a = 7$

$b = 2$

$c = 25$

Igual

$m = 19$

$m = 13$

$m = 7$

$m = 1$

$m = -5$

Digite seu NUSP: 9

Ultimo digito = 9

$m = 28$

$a = 8$

$b = 2$

$c = 28$

Igual

$m = 22$

$m = 16$

$m = 10$

$m = 4$

$m = -2$

2. (valor 2,5 pontos)

Verifique os pedaços de programa abaixo, para calcular x elevado a y , onde x é um real e y um inteiro não negativo. Considere que as variáveis `pot` e `x` são do tipo `double` e `y` e `i` são do tipo `int`. Assuma que 0^0 (zero elevado a zero) é 1 (um). Indique se o trecho está correto (C) ou incorreto (I). Faça um breve comentário ao lado dos trechos que indicar como incorretos, sobre o erro encontrado. Nesta questão, **duas respostas erradas anulam uma certa**. Respostas em branco não serão consideradas erradas.

Exemplo:

```
scanf("%lf %d", &x, &y);           C ( ) I ( x )
pot = 0.0;
for (i=0; i<y; i++)                o resultado é sempre zero.
    pot = pot * x;
printf(" potencia = %g\n", pot);
```

```
2.a)
scanf("%lf %d", &x, &y);           C ( ) I ( x )
pot = 1.0;
for (i=1; i<y; i++)                p/ x elevado a 1,
    pot = pot * x;                 o resultado e' sempre 1
printf(" potencia = %g\n", pot);
```

```
2.b)
scanf("%lf %d", &x, &y);           C ( ) I ( x )
pot = x;                           p/ x elevado a 0,
while (y > 1) {                     o resultado é sempre x
    pot = pot * x;
    y = y - 1;
}
printf(" potencia = %g\n", pot);
```

```
2.c)
scanf("%lf %d", &x, &y);           C ( ) I ( x )
pot = 1.0;
while (y > 1) {                     p/ x elevado a 1,
    y = y - 1;                       o resultado é sempre 1
    pot = pot * x;
}
printf(" potencia = %g\n", pot);
```

```
2.d)
scanf("%lf %d", &x, &y);           C ( x ) I ( )
pot = 1.0;
for (i=y; i>0; i=i-1)
    pot = pot * x;
printf(" potencia = %g\n", pot);
```

```
2.e)
scanf("%lf %d", &x, &y);           C ( ) I ( x )
pot = 1.0;
for (i=y; i>0; i=i-1)              Calcula o fatorial de y
    pot = pot * i;                  ao inves de x elevado a y.
printf(" potencia = %g\n", pot);
```

3. (valor 2,5 pontos)

O programa abaixo tenta resolver o seguinte problema: “Dados $n > 0$ e uma seqüência com n inteiros, verificar se a seqüência está em ordem crescente.”

```
#include <stdio.h>
int main(){
    int n, num, i, numant, ordenado;

    scanf("%d",&n);
    num = 0;
    ordenado = 1;
    for (i=1; i < n; i++){
        numant = num;
        scanf("%d",&num);
        if (num > numant)
            ordenado = 1;
        else
            ordenado = 0;
    }

    if (ordenado == 1)
        printf("A sequencia esta' ordenada\n");
    else
        printf("A sequencia nao esta' ordenada\n");
    return 0;
}
```

O programa está com erro(s) de lógica. Mostre esse(s) erro(s) e reescreva um programa que resolva o problema corretamente (você pode propor a solução que desejar, ela não precisa se parecer com o programa acima). [Se você não consegue explicar qual o erro de lógica o programa tem, basta mostrar uma entrada válida para o problema, mostrando qual seria a saída produzida pelo programa e a saída correta.

SOLUÇÃO:

Há vários problemas de lógica, vamos citar apenas os 2 principais:

- o valor da variável `ordenado` muda a cada comparação, quando deveria mudar para 0 apenas uma única vez (indicador de passagem). Isso faz com que o resultado final seja o resultado do último par, ou seja, para uma seqüência do tipo 1 4 2 5, a saída é "ordenada".
- o valor inicial de `num` é zero. O programa deve ler o primeiro número e não pode assumir que ele é simplesmente zero, caso contrário ele não funciona para números negativos.

```
#include <stdio.h>
int main(){
    int n, num, i, numant, ordenado;
    scanf("%d",&n); /* le o tamanho da seq */
    scanf("%d",&num); /* le o prim num da seq, pois n>0 */
    ordenado = 1;
    for (i=1; i < n; i++){ /* para os demais números */
        numant = num;
        scanf("%d",&num);
        if (num < numant) /* se num < ant a seq nao esta ordenada */
            ordenado = 0;
    }
    if (ordenado == 1)
        printf("A sequencia esta' ordenada\n");
    else
        printf("A sequencia nao esta' ordenada\n");
    return 0;
}
```

4. (valor 2,5 pontos)

Dados $n > 0$ e uma seqüência com n inteiros, determinar o comprimento de um segmento inicial da seqüência de soma máxima, imprimindo também o valor da soma.

Exemplos:

Entrada: $n = 4$

1 2 4 8

Saída: O comprimento é 4 e a soma é 15.

Entrada: $n = 3$

-1 2 -2

Saída: O comprimento é 2 e a soma é 1.

Entrada: $n = 1$

5

Saída: O comprimento é 1 e a soma é 5.

Entrada: $n = 3$

-2 -3 -4

Saída: O comprimento é 0 e a soma é 0.

Entrada: $n = 5$

0 8 -2 -4 6

Possíveis saídas: O comprimento é 2 e a soma é 8.

O comprimento é 5 e a soma é 8.

SOLUÇÃO:

```
int main(){
    int n, num, soma, i, fim, somamax;

    printf(" Digite o numero de numeros da sequencia: \n");
    scanf("%d",&n);

    somamax = 0;
    fim = 0;
    soma = 0;
    for (i=0; i<n; i++){
        printf("Digite o proximo numero da sequencia: \n");
        scanf("%d",&num);
        soma = soma + num;
        if (soma > somamax){
            somamax = soma;
            fim = i + 1;
        }
    }

    printf(" O comprimento e' %d e a soma e' %d\n",fim,somamax);

    return 0;
}
```