

MAC 2166 – Introdução à Computação para Engenharia
ESCOLA POLITÉCNICA – PRIMEIRO SEMESTRE DE 2001
Simulado da Primeira Prova

Na prova, é necessário apresentar um documento original com foto (como RG, carteira da USP, etc).

1 Instruções:

1. Não destaque as folhas deste caderno.
2. Preencha o cabeçalho acima.
3. A prova pode ser feita a lápis. Cuidado com a legibilidade.
4. A prova consta de 4 questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é permitido a consulta a livros, apontamentos ou colegas.
7. Não é permitido o uso de calculadoras.
8. Não é necessário apagar rascunhos no caderno de questões.
9. Qualquer questão pode ser resolvida em qualquer página. Se a resposta não está na página correspondente ao enunciado da questão, indique claramente em qual página se encontra a resposta.

DURAÇÃO DA PROVA: 2 horas

2 Questão 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 n, m, i, k, NUSP;

    scanf("%d", &NUSP); /* digite seu número USP */
    n = NUSP % 10;
    printf("Ultimo digito = %d\n", n);
    if (n > 5)
        n = n - 4;
    else
        if (n < 3)
            n = n + 2;
        else
            n = n + 1;

    printf("n = %d\n", n);

    m = 6;
    i=1;
    while (i < n){
        printf("m = %d\n", m);
        i = i + m % n;
        printf("i = %d\n", i);
        m = 23;
    }

    for (k=2; k < m; k++){
        if (m == 7)
            m=m+2;
        else
            m=m/k;
        if (m > 2 || k < 3)
            printf("m = %d k = %d\n", m, k);
        else
            printf("m = %d\n",m);
        for (i=1; i<k; i++)
            n = n + 1;
    }

    printf("Valores finais: %d %d\n",k,n);
    return 0;
}
```

Para efeito de correção só será considerada a saída do programa. 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.

n	m	i	k	NUSP

saída

3 Questão 2

(valor 2,5 pontos)

Verifique os pedaços de programa abaixo, para cálculo do fatorial de um número natural `num`. Considere que `num`, `fat`, e `i` são variáveis inteiras.

Indique para cada trecho se ele está correto (C) ou incorreto (I), e faça um breve comentário ao lado dos trechos que indicar como incorretos, sobre o erro encontrado. **Nessa questão duas respostas erradas anulam uma certa.** Respostas em branco não são consideradas erradas.

Exemplo:

```
scanf("%d", &num);           C ( ) I ( x )
fat = 0;
while (num > 0) {
    fat = fat * num;         O resultado e' sempre zero.
    num = num - 1;
}
printf("fatorial = %d\n", fat);
```

```
scanf("%d", &num);           C ( ) I ( )
fat = 1;
for (i=1; i < num; i=i+1)
    fat = fat * i;
printf("fatorial = %d\n", fat);
```

```
scanf("%d", &num);           C ( ) I ( )
fat = num;
while (num > 1) {
    num = num - 1;
    fat = fat * num;
}
printf("fatorial = %d\n", fat);
```

```
scanf("%d", &num);           C ( ) I ( )
fat = 1;
while (num > 0) {
    num = num - 1;
    fat = fat * num;
}
printf("fatorial = %d\n", fat);
```

```
scanf("%d", &num);           C ( ) I ( )
fat = 1;
for (i=num; i>0; i=i-1)
    fat = fat * i;
printf("fatorial = %d\n", fat);
```

```
scanf("%d", &num);           C ( ) I ( )
fat = 1;
for (i=num; i>0; i=i-1)
    fat = fat * num;
printf("fatorial = %d\n", fat);
```

4 Questão 3

(valor 2,5 pontos)

O programa abaixo tenta resolver o seguinte problema: 'Dada uma seqüência de reais não-nulos, seguida por um zero, imprimir a sua soma.'

```
#include <stdio.h>
int main(){
    double soma, num;

    soma = 0.0;
    scanf("%lf",&num);
    while (num != 0.0){
        scanf("%lf",&num);
        soma = soma + num;
    }

    printf("Soma = %g\n", soma);
    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).

5 Questão 4

(valor 2,5 pontos)

Uma seqüência a_1, a_2, \dots, a_n de inteiros é uma *progressão geométrica inteira* se existe um inteiro $k \neq 0$ tal que, para todo $1 \leq i < n$, $a_{i+1} = ka_i$.

Escreva um programa em C que, dados um inteiro positivo n e uma seqüência com n inteiros, determina se a seqüência é uma progressão geométrica inteira.

Exemplos:

Entrada: $n = 4$

1 2 4 8

Saída: A seqüência é uma progressão geométrica inteira

Entrada: $n = 3$

0 0 0

Saída: A seqüência é uma progressão geométrica inteira

Entrada: $n = 1$

5

Saída: A seqüência é uma progressão geométrica inteira

Entrada: $n = 3$

-2 2 -2

Saída: A seqüência é uma progressão geométrica inteira

Entrada: $n = 3$

1 0 -1

Saída: A seqüência NÃO é uma progressão geométrica inteira

Entrada: $n = 3$

0 1 2

Saída: A seqüência NÃO é uma progressão geométrica inteira