

MAC2166 – Introdução à Computação para Engenharia
ESCOLA POLITÉCNICA
Segunda Prova – 18 de maio de 2009

Nome: _____

Assinatura: _____

Nº USP: _____ Turma: _____

Professor: _____

Instruções:

1. Não destaque as folhas deste caderno.
2. A prova consta de 3 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	2,5	
2	2,5	
3(a)	1,0	
3(b)	2,0	
3(c)	2,0	
Total	10,0	

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`.

```
# include <stdio.h>

int f1 (int a, int b) {
    int z;

    a = a + b;
    b = a - b;
    z = a + b;

    return z;
}

int f2 (int *a, int b) {
    int z;

    *a = *a + b;
    b = *a - b;
    z = *a + b;

    return z;
}

int f3 (int *a, int b) {

    b = *a + b;
    *a = b + 2;

    return b;
}

int main () {
    int nusp;
    int a, b, c, d, e;
    float f;
    char g;

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

    a = nusp % 5;
    b = a + 2;

    printf ("1: a=%d b=%d\n", a, b);

    c = a;
    d = b;
    e = f1 (c, d);

    printf ("2: a=%d b=%d c=%d d=%d e=%d\n", a, b, c, d, e);

    c = a;
    d = b;
    e = f2 (&c, d);

    printf ("3: a=%d b=%d c=%d d=%d e=%d\n", a, b, c, d, e);

    d = b;
    d = f3 (&d, d);

    printf ("4: a=%d b=%d d=%d\n", a, b, d);

    d = 2 * a + 1;
    b = 2;
    f = d / b;

    printf ("5: a=%d b=%d d=%d f=%f\n", a, b, d, f);

    c = a % 3 + 2;
    g = 'a' + c;

    printf ("6: a=%d b=%d c=%d g=%c\n", a, b, c, g);

    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.

main							
a	b	c	d	e	f	g	nusp

f1		
a	b	z

f2		
a	b	z

f3	
a	b

saída
Entre com o seu no. USP:

2. (vale 2,5 pontos)

Escreva uma função de nome seno que receba como parâmetro um número real x , e devolve uma aproximação para $\text{seno}(x)$ calculada através da seguinte série infinita

$$\text{seno}(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^k \frac{x^{2k+1}}{(2k+1)!} + \dots,$$

incluindo todos os termos até que se encontre um k em que $|\frac{x^{2k+1}}{(2k+1)!}| \leq 10^{-5}$. Este último termo também é incluído na soma.

3. (vale 5,0 pontos)

Numa versão do Exercício-Programa 3 para robôs daltônicos, não usamos luzes verdes nem vermelhas e só usamos luzes amarelas. Um plano de colocação de luzes amarelas é então elaborado. As luzes amarelas serão colocadas nos vértices do polígono regular de n lados inscrito na circunferência de raio 1, sendo o primeiro deles $v_0 = (1, 0)$, e os demais numerados sequencialmente no sentido anti-horário. Os vértices do polígono são denotados então por v_0, v_1, \dots, v_{n-1} . Escreva as funções e programa principal logo abaixo solicitadas. NÃO se deve usar funções da biblioteca matemática.

(a) (vale 1,0 ponto) Escreva uma função com o protótipo

```
void acende_luz (int n, int j, float *x, float *y);
```

que recebe em n o número de vértices e em j o índice da luz a ser acendida e devolve em $(*x, *y)$ a coordenada do vértice v_j . USE a função **seno** da questão anterior, mesmo que não a tenha feito. Caso precise, o valor do **cosseno** DEVE ser calculado como sendo $\text{cosseno}(x) = \text{seno}(\pi/2 - x)$, onde π é 3.14159.

(b) (vale 2,0 pontos) Escreva uma função com o protótipo

```
float mais_distante (int n, float x, float y);
```

que recebe em n o número de luzes, recebe em (x, y) as coordenadas de um ponto e devolve a maior distância de uma das n luzes ao ponto. USE a função do item anterior, mesmo que não a tenha feito. É fornecida (Não a escreva!) e DEVE ser usada a função `dist` com protótipo

```
float dist (float x1, float y1, float x2, float y2);
```

que devolve a distância entre dois pontos (x_1, y_1) e (x_2, y_2) fornecidos.

- (c) (vale 2,0 pontos) Escreva um programa que lê inteiros $n > 0$ e $k > 0$, bem como lê k posições $p_i = (x_i, y_i)$ do robô daltônico no plano. Para cada uma destas posições p_i , o programa deve imprimir a distância d_i da luz amarela mais distante da posição p_i . Ademais, ao final, o seu programa deve imprimir o menor dos valores d_i já impressos anteriormente. Use a função do item anterior, mesmo que não a tenha feito.