MAC 2166 – Introdução à Computação

Engenharia Elétrica e de Computação

Primeiro Semestre de 2022

Segunda Prova – 7 de junho de 2022

Nome do aluno: _____

NUSP: _____ Turma: _____

Assinatura: _____

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 5 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 o uso de artefatos eletrônicos.
7. Não é permitido a consulta a livros, apontamentos ou colegas.
8. Não é necessário apagar rascunhos no caderno de questões.

DURAÇÃO DA PROVA: 2 horas

Questão	Nota
1	
2	
3	
4	
5	
Total	

1. **(2.0 pontos)** Simule a execução do programa abaixo, destacando o que vai ser impresso.

```
#include <stdio.h>

double f1(int x);
int f2(double *y);

int main()
{
  int a, b;
  double c;

  a = 19;
  b = 5;
  c = 11.5;
  printf("a = %d b = %d c = %.2f\n", a, b, c);
  c = f1(a);
  printf("a = %d b = %d c = %.2f\n", a, b, c);
  a = 19;
  b = 5;
  c = 11.5;
  b = f2(&c);
  printf("a = %d b = %d c = %.2f\n", a, b, c);
  a = 19;
  b = 5;
  c = 11.5;
  c = f1((int) c);
  printf("a = %d b = %d c = %.2f\n", a, b, c);
  return 0;
}

double f1(int x) {
  x = x / 2;
  printf ("Em f1: x = %d\n", x);
  return ((double) x / 2);
}

int f2(double *y) {
  *y = *y / 2;
  printf("Em f2: y = %.2f\n", *y);
  return ((int) *y );
}
```

As próximas questões tratam de lançamento oblíquo. Considere um novo jogo do Bixusp, em que os times dispõem de um "canhão" que lança bolinhas de gude em lançamento oblíquo, e o time ganha ponto se acertar em um copo colocado no chão a uma certa distância.

Lembre que no lançamento oblíquo podemos considerar que o projétil descreve um movimento retilíneo uniformemente variado (MRUV) no eixo y e um movimento retilíneo uniforme (MRU) no eixo x. Assim, se soubermos a velocidade v_0 e o ângulo θ em que o projétil é lançado, podemos calcular a projeção da velocidade nos dois eixos, e, com isso, o tempo que ele leva para atingir o solo novamente, a distância que o projétil percorre até cair, etc.

No MRUV, sabemos que a velocidade v do objeto no instante t é dada por $v = v_0 + at$, onde v_0 é a velocidade do objeto no instante $t = 0$ e a é a aceleração do objeto. No MRU a posição do objeto no instante t é dada por $r = r_0 + vt$, onde r_0 é a posição do objeto no instante $t = 0$ e v é a velocidade do objeto.

Essa questão terá 4 partes, que devem ser resolvidas independentemente. Você pode usar uma função descrita num item nos demais, mesmo que você não a tenha feito. Você pode assumir que a aceleração da gravidade na USP é $g = 9.8\,\mathrm{m/s^2}$ e você pode usar 3.1415 para o valor de π.

2. **(3.0 pontos)** Faça uma função de protótipo

```
double cosseno(double x, double epsilon);
```

que recebe um real x e devolve uma aproximação de $\cos x$ dada pela série infinita

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots + (-1)^k \frac{x^{2k}}{(2k)!} + \ldots$$

incluindo todas as parcelas da série até a primeira parcela com valor absoluto menor que ε.

3. **(1.0 ponto)** Faça uma função de protótipo

```
double tempo(double v0, double theta);
```

que recebe a velocidade e ângulo de lançamento de um projétil num lançamento oblíquo, e calcula o tempo que o projétil leva para voltar ao chão. Lembre que o valor da projeção da velocidade no eixo y é dado por $v_0 \cos(\frac{\pi}{2} - \theta)$, o projétil descreve um MRUV no eixo y, e o tempo total que ele permanece no ar é duas vezes o tempo até a altura máxima (ponto de velocidade zero). Para calcular cossenos, você deverá usar a função da questão anterior, mesmo que você não a tenha feito (você pode usar um valor de ε que você achar conveniente).

4. **(1.0 ponto)** Faça uma função de protótipo

```
double distancia(double v0, double theta);
```

que recebe a velocidade e ângulo de lançamento de um projétil num lançamento oblíquo, e calcula a distância que o projétil vai percorrer até cair. Lembre que o valor da projeção da velocidade no eixo x é dado por $v_0 \cos\theta$, e o projétil, no eixo x, descreve um MRU. Você deverá usar a função do item anterior para calcular o tempo que o projétil fica no ar, mesmo que não a tenha feito.

5. **(3.0 pontos)** Faça um programa que leia v_0, o valor da velocidade com que o canhão lança a bolinha de gude, a distância d até o copo, e o raio r do copo, e que imprima o ângulo θ em que o lançamento deve ser feito para fazer um ponto no jogo.

   Para fazer essa questão, **você não se lembra se existe uma fórmula fechada para o ângulo pedido**, e deve escrever um programa que encontra o ângulo por aproximações sucessivas. Note que se o lançamento for feito com $\theta = \frac{\pi}{2}$, a bolinha cai na sua cabeça, e lembre que a distância máxima é atingida quando o ângulo de lançamento é $\frac{\pi}{4}$. Assim, sua primeira tentativa poderia ser o ângulo no meio do intervalo $[\frac{\pi}{4}, \frac{\pi}{2}]$, isto é, $(\frac{\pi}{4}+\frac{\pi}{2})/2 = 3\pi/8$. Se o tiro for muito curto, você pode atualizar seu intervalo para o intervalo $[\frac{\pi}{4}, \frac{3\pi}{8}]$. Se for muito longo, seu novo intervalo seria $[\frac{3\pi}{8}, \frac{\pi}{2}]$. Você segue pegando o meio do intervalo até que a diferença entre a distância do copo e a posição em que a bolinha cai é menor que o raio do copo. Se o copo estiver a uma distância maior do que é possível atingir com ângulo de lançamento $\frac{\pi}{4}$, seu programa deverá informar isso.

   Você deve usar a função `distancia()` da Questão 4 em seu programa, mesmo que você não a tenha feito.