

MC102: Algoritmos e Programação de Computadores

Prof. Juliana de Santi - Turmas A e B

2ª Prova - 08/05/2012

Nome:

RA:

Turma:

Questão	Valor	Nota
1	3,0	
2	3,0	
3	2,0	
4	2,0	
Total	10,0	

Instruções: A duração da prova é de 120 minutos. **Não é permitida consulta** a qualquer material. Em caso de fraude, todos os envolvidos receberão nota zero. *Somente serão consideradas respostas nos espaços marcados.* Use os versos das páginas como rascunho. Você pode fazer a prova a lápis (desde que o resultado final seja legível). Boa prova!

1. **(3.0)** Considere o código em C abaixo (assuma que no lugar de D seja usado o último dígito de seu RA):

```
#include <stdio.h>

void soma1(int *q, int c);
int soma2(int ra);

int i = 100;
int j = 200;

int main(){
    int i,k,ra,*p, x;
    p = &x;
    ra = D; //Use o último dígito de seu RA.

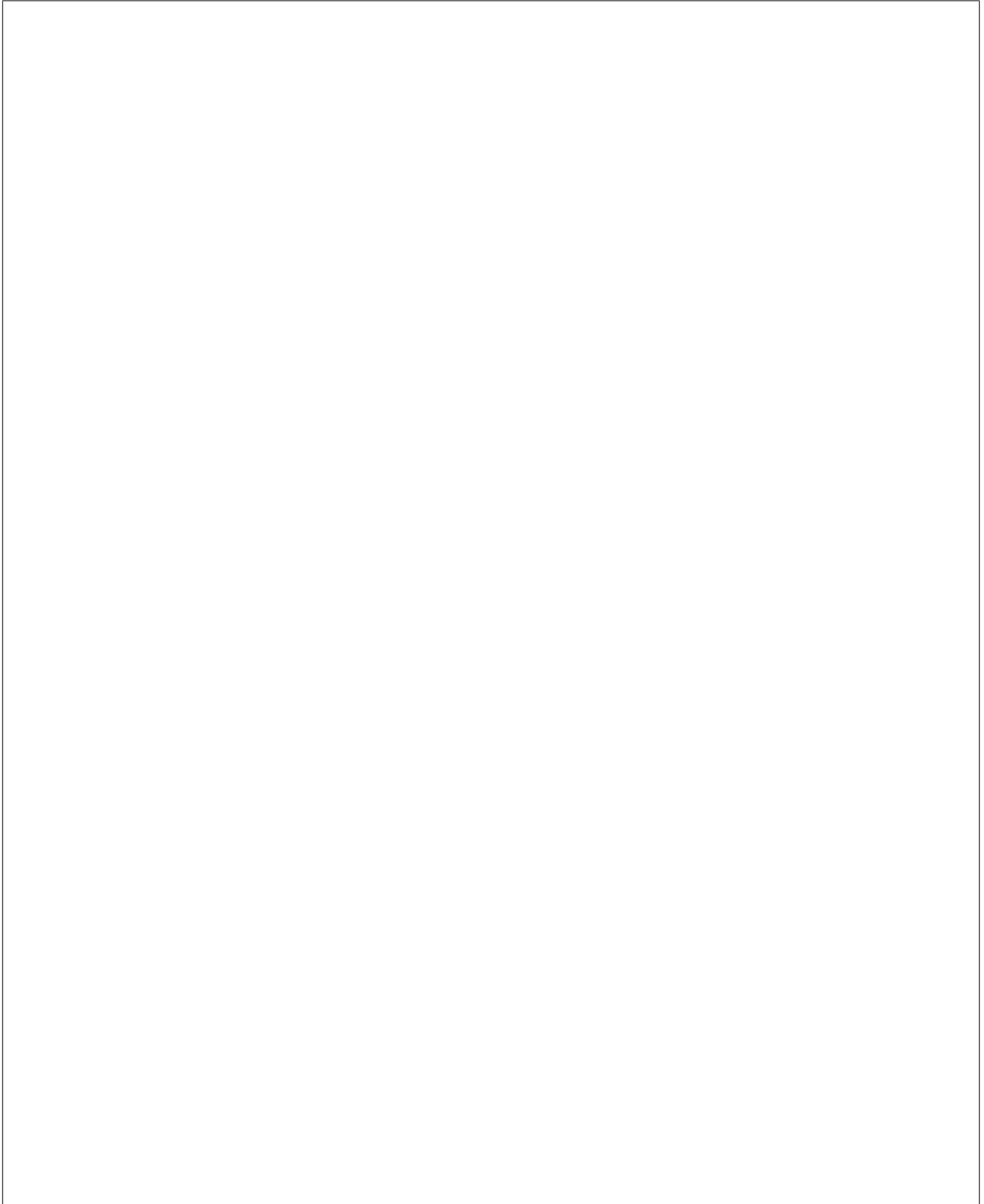
    for(i = 0; i < 3; i++){
        *p = 10*i;
        soma1(&ra, *p);
        k = soma2(ra);
        printf("%d, %d\n",ra, k);
    }
}

void soma1(int *q, int c){
    int soma = *q+i+c;
    *q = soma;
}

int soma2(int ra){
    int k = j;
    ra = ra + k;
    return(ra);
}
```

- (a) **(1.5)** Determine quais são as variáveis locais e globais deste programa, identificando a que função pertence cada variável local.

- (b) **(1.5)** Mostre o que será impresso na tela do computador quando for executado este programa (lembre-se de usar o último dígito de seu RA no lugar de D).



2. (3.0) Um inteiro positivo n é **pitagórico** se existem inteiros positivos a e b tais que $a^2 + b^2 = n$. Por exemplo, 13 é pitagórico pois $2^2 + 3^2 = 13$.

(a) (1.5) Escreva uma função `int Pit(int n, int *a, int *b)` que dado um inteiro positivo n verifica se n é pitagórico. Se for, então `Pit` deve devolver 1 e os valores a e b nos parâmetros correspondentes. Caso contrário, `Pit` deve devolver 0. **Dica:** Use o seguinte trecho de código na sua solução:

```
for(i=1; i<= sqrt(n); i++)
    for(j=1; j<= sqrt(n); j++){
        //teste se i e j são soluções
    }
```

- (b) **(1.5)** Use a função anterior e escreva um programa que lê um inteiro n e imprime inteiros a e b tais que $a^2 + b^2 = n$ (se existirem) ou então imprime a mensagem *O número lido não é pitagórico*.

3. (2.0) Escreva uma função que recebe como parâmetro uma matriz quadrada de floats de tamanho no máximo 30×30 , um inteiro n que indica o tamanho da matriz, e três floats passados por referência (**void fun(float mat[30][30], int n, float *media, float *min, float *max)**). A função deve devolver em **media**, a média dos valores da matriz, em **min** o menor valor na matriz, e em **max**, o maior valor na matriz.

4. (2.0) Uma palavra (string) é um **anagrama** de outra palavra se ambas têm as mesmas letras na mesma quantidade. Por exemplo, ABCBD e BADCB são anagramas entre si. Suponha nesta questão que as únicas letras que aparecem nas palavras são as letras maiúsculas 'A', 'B', ..., 'Z' (inclui 'K', 'Y' e 'W' também). Escreva uma função `int Anagrama(char s[], char r[], int n)` que recebe duas strings com n caracteres (ou seja, `s[n] == r[n] == '\0'`) e devolve 1 se `s` é um anagrama de `r` e devolve 0 caso contrário. **Dica:** você pode usar um vetor `int freq[26]` para computar a frequência de cada letra em uma string. Em `freq[0]` deve ser guardado o número de vezes que 'A' aparece na string, em `freq[1]` o número de vezes que 'B' aparece etc. Se `char letra` é uma variável que guarda uma letra (maiúscula) então a posição correspondente a esta no vetor `freq[]` é dado por `(letra-'A')`.