



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Projeto e Desenvolvimento de Algoritmos
AP2 2º semestre de 2005.

Nome –

Assinatura –

Questão 1 (2.5 pontos)

Um programador de computadores mora em uma rua com n casas e de frente para o mar. Portanto, somente há casas de um lado da rua. As casas são numeradas consecutivamente, começando em 1. Toda noite o programador sai de casa para passear com seu cachorro e aleatoriamente escolhe um lado para percorrer. Ele sempre vai até o final da rua e volta para casa. Uma noite ele decidiu somar os números das casas por onde passou. Esta soma foi feita somente no caminho de ida e excluía a sua própria casa. Na noite seguinte ele tomou a direção contrária e fez a mesma conta. Para sua enorme surpresa as duas somas deram o mesmo resultado. Embora isto seja, obviamente, determinado pelo número de sua casa e do número de casas de sua rua, ele acha que isto é um sinal de sorte e decide que a partir de agora somente irá morar em casas com esta propriedade, que ele chamou de “casa sortuda”.

Um mês após sua descoberta, nosso programador se vê forçado a se mudar para uma outra rua, também de frente para o mar, com 57.121 casas onde todos os imóveis estão disponíveis para compra. Escreva um algoritmo que determine o(s) número(s) da(s) casa(s) que ele pode comprar para satisfazer sua decisão a respeito de “casas sortudas”

constantes

DIM = 57121

início

para $i \leftarrow$ até DIM-1 faça

somaAntes \leftarrow 0

somaDepois \leftarrow 0

para $j \leftarrow 1$ até $i-1$ faça

somaAntes \leftarrow somaAntes + j

próximo j

para $j \leftarrow i+1$ até DIM faça

somaDepois \leftarrow somaDepois + j

próximo j

se somaAntes=somaDepois então

```

        imprima 'Voce pode comprar a casa: ', i
    fim se
    próximo i
fim

```

Questão 2 (2.5 pontos)

Nosso amigo Sr. Zé Sá anda com pouca sorte. Outra noite de ventos e tempestade arrancou novamente os telhados e portas das baias do haras Mensalão. Os cavalos, todos de raça e de alto valor, são alojados em 50 baias contíguas, numeradas de 1 a 50. Felizmente, nesta noite, alguns cavalos estavam viajando para participar de uma competição, e as baias não estavam todas ocupadas.

Dessa vez, o Sr. Zé Sá encontrou outra maneira de resolver o problema. Para evitar que os 18 cavalos que dormiam no haras fiquem desprotegidos, o Sr. Zé Sá pode encomendar ao seu carpinteiro no máximo duas pranchas de madeira de qualquer comprimento. O Sr. Zé Sá não pode mudar os cavalos de baias, uma vez que eles são bastante sensíveis e a mudança prejudica o seu desempenho em competições. Escreva um algoritmo que determine o número de baias protegidas (ocupadas ou não) de modo que todos os cavalos sejam protegidos e o comprimento total das tábuas seja o menor possível.

O algoritmo deve ler as posições das baias ocupadas pelos 18 cavalos em ordem crescente.

Exemplo:

```

3
4
6
8
14
15
16
17
21
25
26
27
30
31
40
41
42
43

```

Numero de baias protegidas: 33

início

```

    para i ← 1 até 18 faça
        leia baiaOcupada[i]
    próximo i

```

```

    maiorIntervalo ← 0

```

```

    para i ← 2 até 18 faça
        intervalo ← baiaOcupada[i] - baiaOcupada[i-1] - 1
        se intervalo > maiorIntervalo então
            maiorIntervalo ← intervalo
    fim se
    próximo i

```

```

    baiasProtegidas ← baiaOcupada[18] - baiaOcupada[1] + 1 -
                        maiorIntervalo;

    imprima 'Numero de baias protegidas: ', baiasProtegidas
fim

```

Questão 3 (2.5 pontos)

O sítio www.jogueaqui.com.br mantém uma série de jogos on-line disponíveis para os seus associados. Ao final do jogo o usuário é informado de sua pontuação e do ranking dos 10 melhores resultados até aquela data. Escreva um algoritmo que leia os pontos do usuário, o ranking dos melhores resultados (não ordenado) e determine o novo ranking com a inclusão ou não dos pontos do usuário.

Exemplo (os caracteres em **negrito** correspondem às entradas fornecidas pelo usuário, os demais, saídas geradas pelo computador):

numero de pontos: **1200**

ranking atual

1120

1350

1080

1960

1085

1201

998

1398

2198

1007

novo ranking

1120

1350

1080

1960

1085

1201

1200

1398

2198

1007

```

início
    imprima 'numero de pontos: '
    leia nPontos
    imprima 'ranking atual'

    para i ← 1 até 10 faça
        leia pontos[i]
    próximo i

    pos ← 1

    para i ← 2 até 10 faça
        se pontos[i] < pontos[pos] então
            pos ← i
        fim se
    próximo i

```

```

se nPontos > pontos[pos] então
    pontos[pos] ← nPontos
fim se

imprima 'novo ranking'
para i ← 1 até 10 faça
    imprima pontos[i]
próximo i
fim

```

Questão 4 (2.5 pontos)

Escreva um algoritmo para determinar o número de arranjos simples possíveis para m elementos tomados p a p . A fórmula de cálculo deste valor é:

$$A_m^p = \frac{m!}{(m-p)!}$$

Para facilitar a construção do seu algoritmo, escreva uma função *fatorial* que retorna o fatorial de um número inteiro passado como argumento. Lembre-se que o fatorial de um número inteiro n pode ser calculado como:

$$n! = 1*2*3*...*n$$

Exemplo:

$m = 5$

$p = 3$

$A(5, 3) = 60$

Observação: Foram mostrados em **negrito** os valores digitados pelo usuário.

```

início
    imprima 'm = '
    leia m
    imprima 'p = '
    leia p
    imprima 'A(', m, ', ', ' ', p, ') = ',
        fatorial(m)/fatorial(m-p)
fim

```

```

função fatorial(entradas: n)
início
    resultado ← 1
    para i ← 1 até n faça
        resultado ← resultado * i
    próximo i
fim

```