

Introdução à Programação (CC111)

2011/2012

1º Teste Escrito (23.11.2011)

duração: 2 horas + 15 minutos

(Cotação: 1×3, 2, 1.5, 5, 1, 1, 1, 1, 1, 3.5)

N.º Nome

1. Em cada uma das alíneas, diga o que escreve cada fragmento de código dado, supondo que as variáveis `x`, `y` e `z` são tipo `int` e têm inicialmente os valores 6, 19, e 21.

a) <input type="text"/>	b) <input type="text"/>	c) <input type="text"/>
<pre>x = z; z = y; y = x; if (y == z) x = 1; else if (y > z) x = 2; printf("x = %d", x);</pre>	<pre>for(y = z; y > x; y = y-5) z = y - 2*x; printf("%d %d %d", x, y, z);</pre>	<pre>while(--y > z-x) { x--; z += 2; } printf("%d %d", y, z);</pre>

2. Sendo $h(x) = 2(x+10)$ se $x \leq 0$ e $h(x) = -x + 5$ se $x > 0$, o programa seguinte deveria imprimir o valor de $h(x)$ para um inteiro x lido da entrada padrão. Porém tem alguns erros. Apresente uma versão corrigida (sem alterar o número de variáveis nem de `printf`'s) e explique as correções.

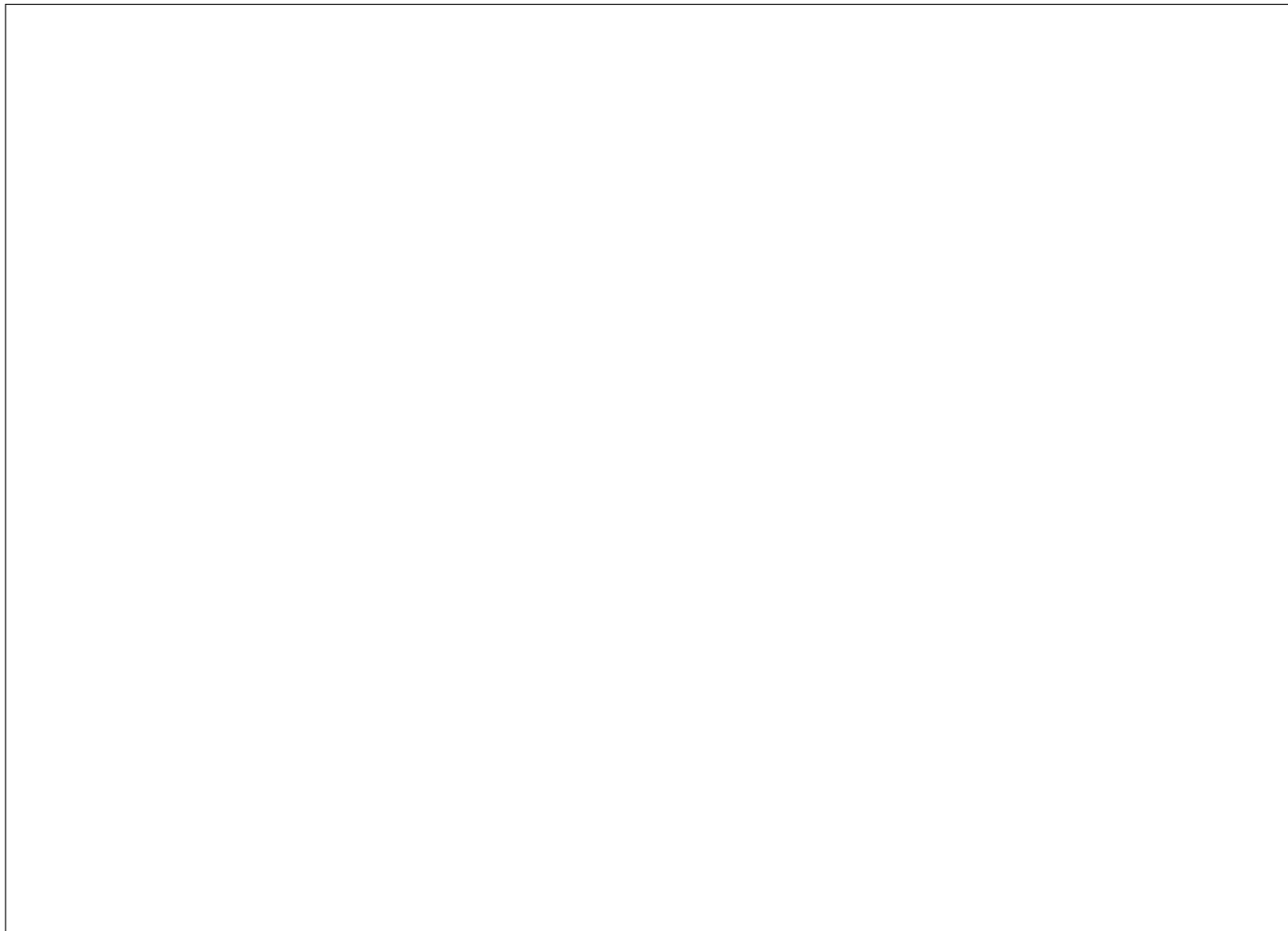
```
#include <stdio.h>
int main(){
    scanf("%d", x)
    if (x > 0)
        x = -(x+5);
    if (x <= 0)
        x = 2*x+10;
    printf("%d", x);
}
```

3. Num programa com duas variáveis inteiras `a` e `b`, diga, justificando, o que é escrito na saída padrão pela sequência de instruções `a = 5; b = 3; calcula(a,b); printf("(a,b) = (%d,%d)\n", a,b);`; se a função `calcula` for definida por:

```
void calcula(int x, int y) {
    x = (y*4-x)/5;
    y = y*y*y*y;
    printf("(x,y) = (%d,%d)\n", x,y);
}
```

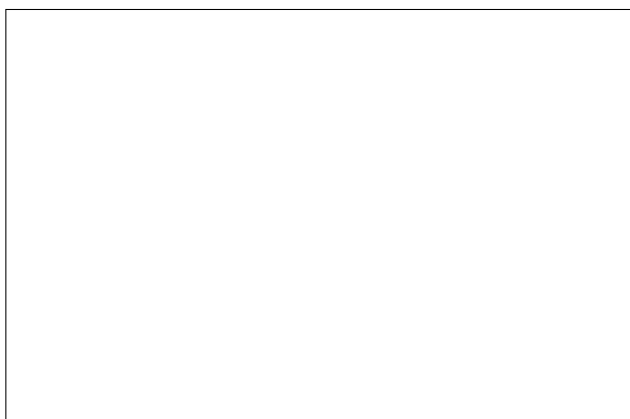
(CONTINUA, v.p.f.)

4. Pretende-se analisar n sequências de inteiros terminadas por -1 e, para cada uma, imprimir o número de ordem das posições das duas primeiras ocorrências de inteiros menores do que o seu primeiro valor e ímpares. Escreva um programa para resolver o problema sem usar variáveis indexadas. Os dados são lidos da entrada padrão e n é o primeiro inteiro lido. O número de ordem do primeiro valor de cada sequência é 1 e o terminador -1 não pertence à sequência. Na saída, $*$ indica inexistência. Se der 4 7 -2 8 5 6 3 20 0 -1 2 1 5 1 0 -1 2 0 3 5 -1 9 -6 13 -4 7 0 12 -1, o programa escreveria quatro linhas (com 4 6, 2 4, * *, e 5 *, respectivamente). Apresente uma interpretação informal para cada variável que usar e comentários para explicar os passos principais.

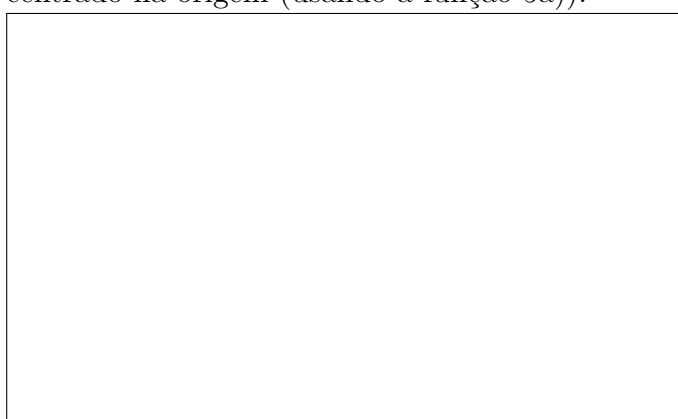


5. Sendo P um ponto com coordenadas (x, y) num referencial ortonormado, a distância euclidiana de P à origem do referencial é dada por $\sqrt{x^2 + y^2}$. Implemente:

a) `int quadrado_de_dist(int x, int y)`
para calcular o quadrado da distância de um ponto (x, y) à origem.



b) `int conta(int pontos[][2], int n, int r)`
para determinar quantos dos n pontos dados por `pontos` pertencem ao exterior do círculo de raio r centrado na origem (usando a função 5a)).



(CONTINUA, v.p.f.)

6. Considere os seguintes fragmentos de código supondo que as n primeiras posições de um vector dado por x guardam uma sequência de inteiros ordenada por ordem estritamente decrescente, sendo $1 \leq n \leq 2^{15} = 32768$, e que todas as outras variáveis têm tipo `int`.

// Caso A

```
a = 0; b = n-1; t = (n-1)/2;
```

```
while (b >= a && x[t] != v) {
```

```
    if (v > x[t]) b = t-1;
```

```
    else a = t+1;
```

```
    t = (a+b)/2;
```

```
}
```

```
if (b >= a) printf("%d",t);
```

// Caso B

```
i=0;
```

```
while(i < n && x[i] != v)
```

```
    i++;
```

```
if (i < n)
```

```
    printf("%d",i);
```

a) Que significado tem o valor escrito em cada caso? Justifique.

b) Se $n = 65$ e $v < x[n-1]$, quantas vezes é testada a condição que controla o ciclo `while`, em cada caso? Justifique.

c) Identifique as abordagens que estes excertos de programa implementam e indique as vantagens e desvantagens de uma em relação à outra, em termos de eficiência e condições de aplicabilidade.

7. Uma empresa comercializa 40 produtos e identifica-os com códigos de 1 a 40. Em cada dia, recebe uma sequência de encomendas que processa por ordem de chegada. Cada encomenda envolve apenas um produto, mas pode requerer o envio de mais do que uma unidade desse produto. Durante o dia não haverá reposição de stocks e as encomendas que não puderem ser completamente satisfeitas são perdidas (e nenhuma unidade de produto será enviada). Pretende-se escrever um programa para processar duas sequências de pares de inteiros, ambas terminando por 0 0. A primeira representa o stock existente no início de um dia, sendo apenas mencionados os pares $p \ s$ dos produtos p para os quais o stock s não é nulo. A segunda representa as encomendas recebidas nesse dia, por ordem de chegada, e no formato $p \ q$, sendo $q \geq 1$ o número de unidades do produto p que o cliente pretende. O programa imprime o número total de unidades que a empresa não vendeu por não poder satisfazer algumas encomendas e o número total de unidades que foram vendidas (sem distinguir produtos). Implemente o programa e introduza comentários sucintos para explicar ideias ou passos principais. Terá de incluir e completar o código seguinte.

```
int main() {
    int stock[NUMCODIGOS+1], balanco[2];
    stock_existente(stock); // ler dados sobre stock e inicializar stock de acordo
    despachar_encomendas(stock,balanco); // na saída, terá em balanco os dois números pedidos
    printf("%d no total de %d\n",balanco[0],balanco[1]); // é possível 0 no total de 0
    return 0;
}
```