

Programação Imperativa 2022/2023 (CC1003), DCC/FCUP

Folha 2

2.1 Os seguintes fragmentos de programas ilustram o resultado de operadores lógicos; indique o resultado de cada um, assumindo que *i*, *j*, *k* são variáveis inteiras.

```
1. i = 10; j = 5;
   printf("%d", i < j);
   1 (verdadeiro)      -10 < 5
```

muda o sinal do número

```
2. i = 2; j = 1;
   printf("%d", !!i + !j);
   1 (resultado conta) 2 + (-1)
```

③

```
i = 5; j = 0; k = -5;
printf("%d", i && j || k);
1          (5 ^ 0) v (-5)
```

```
4. i = 1; j = 2; k = 3;
   printf("%d", i < j || k);
   1          (1 < 2) v 3
```

```
5. i = 3; j = 4; k = 5;
   printf("%d", i < j || ++j < k); (3 < 4) v (5 < 5)
   printf("%d %d %d", i, j, k);
   1 3 4 5
```

6.

```
i = 7; j = 8; k = 9;
printf("%d", i - 7 && j++ < k); (7-7) ^ (8 < 9)
printf("%d %d %d", i, j, k);
0 7 9
↑
```

2.2 As condições expressas nos seguintes fragmentos de programa não surtem o efeito indicado nas mensagens. Para cada um deles, simule a execução e indique o resultado. Em seguida, corrija as condições para que correspondam à intenção expressa na mensagem.

```
1. int n = 0;
   if (1 <= n <= 10)
       printf("n entre 1 e 10\n");
```

```
2. int i = 1;
   if (i == 2 || 3)
       printf("i igual 2 ou 3\n");
   (i==2) || (i==3)
```

```
3. int i = 2, j = 1, k = 0;
   if (i==j==k)
       printf("i,j,k iguais\n");
   (i==j) && (j==k)
```

```
4. int i = 1, j = 2, k = 3;
   if (i!=(j&k))
       printf("i diferente j e k\n");
   (i!=j) && (i!=k)
```

2.3 (Plataforma codex) Escreva um programa `maior.c` que lê três valores inteiros da entrada padrão e imprime o maior destes valores. Para submeter ao sistema de testes automáticos, o programa deve imprimir apenas o maior valor e uma mudança de linha `\n`; não deve imprimir mensagens extra. Eis um “esqueleto” de uma solução:

```
... //diretivas include
int main(){
    int a, b, c, maior;
    scanf("%d %d %d", &a, &b, &c); // ler 3 valores
    ...                          // determinar o maior
    printf("%d\n", maior);        // imprimir o resultado
}
```

```
if a > b;
    maior = a
elif
    maior = b
if c > maior;
    maior = c
```

2.4 Escreva um programa que lê uma quantia inteira de euros e mostra como pagar essa quantia em notas de 20 euros, 10 euros, 5 euros e moedas de 1 euro. Exemplo:

```
Quantia em EUR? 93
notas EUR 20: 4
notas EUR 10: 1
notas EUR 5: 0
moedas EUR 1: 3
```

2.5 Escreva um programa que lê 3 valores inteiros da entrada-padrão e escreva a *mediana*, isto é, o valor no meio quando os colocamos por ordem crescente. Assim, se os valores forem a, b, c com $a \leq b \leq c$, então a mediana será b . Exemplo:

```
Primeiro valor: 7
Segundo valor: 12
Terceiro valor: 9
Mediana: 9
```

2.6 (Plataforma codex) Baseando-se na solução do exercício 2.5, complete a definição da seguinte função para calcular a mediana de 3 inteiros a, b e c .

```
int mediana(int a, int b, int c){
    /* calcular e retornar a mediana de a, b, c */
}
```

2.7 (Plataforma codex) A linguagem C não tem um operador para potências, mas podemos calcular x^n efetuando multiplicações sucessivas (assumindo $n \geq 0$ inteiro). Complete a definição da seguinte função para calcular potências.

```
int potencia(int x, int n){
    /* calcular e retornar  $x^n$  */
}
```

2.8 Escreva um programa que lê uma sequência de inteiros do *standard input* terminada pelo valor 0 e imprime a sua *média aritmética*. Admita que é dado pelo menos um valor além de zero. Exemplo:

```
10 3 5 -20 6 0
0.8
```

2.9 Escreva um programa que lê um inteiro > 1 e imprime a lista de fatores primos. Exemplo:

```
Numero inteiro: 24
24: 2 2 2 3
```

Sugestão: utilize um ciclo em que faz divisões sucessivas por inteiros crescentes começando por 2, depois 3, etc. Note que se dividir tantas vezes quanto possível por um inteiro já não será possível dividir pelos seus múltiplos, logo vamos apenas obter os fatores primos tal como pretendido.

2.10 Um ano é *bissexto* se for divisível por 4, exceto se for também divisível por 100 e não for divisível por 400. Por exemplo: 1980 foi bissexto porque 1980 é divisível por 4 mas não por 100; 1700 não foi bissexto porque, apesar de ser divisível por 4, também é divisível por 100 e não é por 400. Defina uma função `int bissexto(int n)` que verifica esta condição para um ano `n`; o resultado deve ser 1 se o ano for bissexto e 0 caso contrário.

2.11 (Plataforma codex) Escreva uma função `int prox_bissexto(int n)` que calcula o próximo ano bissexto a partir do ano dado; se `n` for um ano bissexto, então o resultado deve ser `n`. Exemplos: `prox_bissexto(1980)` retorna 1980; `prox_bissexto(2017)` retorna 2020.

2.12 Escreva um programa que leia da entrada padrão uma data (i.e., 3 inteiros que representam dia, mês, ano) e determine a data do dia seguinte.