

Exame de Programação Imperativa

17 Janeiro de 2020

Q1: Somar os divisores de um número

Cotação: 15 pts.

Escreva uma definição de uma função `soma_divs` com um argumento inteiro positivo n e que calcula a soma dos divisores positivos de n inferiores a n .

Exemplo: os divisores de 12 são 1, 2, 3, 4, 6 e 12; logo `soma_divs(12)` deve retornar 16 ($= 1 + 2 + 3 + 4 + 6$).

Neste exercício deve submeter **apenas a definição da função** `soma_divs`.

Q2: Sequências de caracteres repetidos

Cotação: 20 pts.

Escreva um programa completo que lê caracteres um de cada vez da entrada-padrão até ao fim do ficheiro (EOF) e calcular e imprime *o maior comprimento duma sequência de caracteres iguais*.

Exemplo: se a entrada for

AABBBAACCCBB

então a resposta deve ser 3 (correspondente à sequência **BBB**). Note que estamos apenas interessado em *sequências* de caracteres repetidos, logo os dois últimos caracteres **B** não somam aos três anteriores.

Nota: apesar do exemplo conter apenas letras, o seu programa deve funcionar também para outros caracteres (incluindo mudanças de linhas e espaços); não há necessidade de tratar letras de forma especial.

Neste exercício deve submeter um **programa completo** i.e., incluindo a função `main`. Pode usar funções auxiliares se quiser.

Q3: Converter um numeral binário num inteiro

Cotação: 15 pts.

Recorde que na representação em binário usamos apenas algarismos 0 e 1. A representação é posicional com base 2: o algarismo mais à esquerda (menos significativo) vale exatamente o seu valor; as posições seguintes valem 2, 4, 8, etc., sempre em potências de 2.

Exemplo: 1101 representa $2^3 + 2^2 + 2^0 = 8 + 4 + 1 = 13$.

Escreva a definição de uma função `binário` que converte uma cadeia de caracteres com algarismos binários no valor inteiro correspondente. Os algarismos da cadeia estão pela ordem normal (ou seja, do mais significativo para o menos significativo) Pode assumir que a cadeia só contém caracteres '0' e '1' e está terminada por '\0'; além disso, o resultado será sempre inferior a `INT_MAX` (logo é representável num `int`).

Neste exercício deve submeter **apenas a definição da função binario**.

Pode testar a sua função com alguns casos antes de submeter. Exemplos:

```
assert(binario("1") == 1);
assert(binario("10") == 2);
assert(binario("110") == 6);
assert(binario("1000") == 8);
assert(binario("1111") == 15);
```

Q4: Encontrar k -ésimo menor valor numa sequência

Cotação: 15 pts.

Dada uma sequência de n valores $v[0], v[1], \dots, v[n-1]$ não ordenados, pretendemos encontrar o k -ésimo menor valor:

- se $k = 0$, o resultado deve ser o menor valor da sequência;
- se $k = 1$, o resultado deve ser o 2º menor valor;
- se $k = 2$, o resultado deve ser o 3º menor valor;
- e assim sucessivamente até $k = n - 1$ (que será o maior valor).

Exemplo: se v for $\{17, 11, 30, 17, 40\}$ então: o 0-ésimo valor é 11; o 1-ésimo valor e 2-ésimo valores são ambos 17; o 3-ésimo valor é 30; e o 4-ésimo valor é 40.

Podemos resolver o problema de encontrar o k -ésimo menor valor ordenando a sequência de valores: o k -ésimo menor valor fica então na posição k da sequência ordenada.

Implemente uma função

```
int kmin(int k, int v[], int n);
```

que retorna o k -ésimo menor valor do vector v ; n é o número de elementos do vector; pode ordenar o vector usando um qualquer algoritmo de ordenação estudado nas aulas.

Indique num comentário qual o algoritmo que implementou.

Neste exercício deve submeter **apenas a definição da função** `kmin`.

Q5: Questões de escolha múltipla

Cotação: 15 pts.

- Para cada uma das seguintes questões apenas uma das alternativas é correta
- Não responda se acha que não sabe: **respostas erradas descontam metade da cotação de uma resposta certa**
- Não obterá “feedback” das respostas corretas durante o exame; o resultado será sempre *Accepted*
- Pode responder múltiplas vezes; apenas a última submissão será cotada

Num programa em C, a função `main`:

- (a) deve terminar com `return 0` para indicar que não ocorreu um erro
- (b) deve terminar com `return 0` para indicar que ocorreu um erro
- (c) não pode conter a instrução `return`

As fases da tradução de um programa C para código-máquina são:

- (a) pré-processamento, compilação e execução do ficheiro binário
- (b) pré-processamento, compilação e ligação do ficheiro binário

(c) compilação, ligação e execução do ficheiro binário

Na linguagem C qual é o resultado da expressão $11\%3+10/2$?

- (a) 7
- (b) 7.5
- (c) 8

Na linguagem C qual é o resultado da expressão $11/3+10\%2$?

- (a) 3
- (b) 3.666666
- (c) 4

Qual das seguintes expressões lógicas verifica se uma variável `c` de tipo `char` representa uma letra maiúscula?

- (a) `'A' <= c <= 'Z'`
- (b) `c <= 'A' || c >= 'Z'`
- (c) `c >= 'A' && c <= 'Z'`

Qual das seguintes expressões lógicas verifica se o valor de uma variável `n` de tipo `int` está no intervalo -10 a 10?

- (a) `-10 <= n <= 10`
- (b) `n <= 10 || n >= (-10)`
- (c) `n <= 10 && n >= (-10)`

O seguinte fragmento de programa imprime alguns inteiros dentro do intervalo de 1 a 100.

```
for(int k = 1; k <= 100; k++) {  
    if(!(k%2 == 0 && k%3 == 0))
```

```
    printf("%d\n", k);  
}
```

Indique qual das opções descreve o subconjunto dos valores impressos.

- (a) os inteiros que são múltiplos de 2 e de 3
- (b) os inteiros que não são múltiplos de 2 nem de 3
- (c) os inteiros que não são múltiplos de 2 ou não são múltiplos de 3

O seguinte fragmento de programa imprime alguns inteiros dentro do intervalo de 1 a 100.

```
for(int k = 1; k <= 100; k++) {  
    if(k%2 == 0 || k%3 == 0)  
        printf("%d\n", k);  
}
```

Indique qual das opções descreve o subconjunto dos valores impressos.

- (a) os inteiros que são múltiplos de 2 e de 3
- (b) os inteiros que não são múltiplos de 2 nem de 3
- (c) os inteiros que são múltiplos de 2 ou de 3

Qual dos seguinte ciclos **não é equivalente** aos outros dois (assumindo que a instrução omitida (...) é sempre igual)?

- (a) `i = 0;`
`while (i<100) { ...; i++; }`
- (b) `i = 0;`
`do { ...; i++; } while (i<100);`
- (c) `for(i=0; i<100; i++) { ...; i++; }`

Qual dos seguinte ciclos **não é equivalente** aos outros dois (assumindo que a instrução omitida (...) é sempre igual)?

- (a) `i = 1;`
`while (i<100) { ...; i++; }`
- (b) `i = 0;`
`do { i++; ...; } while (i<100);`

(c) `for(i=1; i<=100; i++) { ...; }`

Considere a seguinte expressão usando a função `rand()` da biblioteca-padrão:

```
int r = rand()%10 - 5;
```

Qual o intervalo de valores que poderá tomar o resultado `r`?

- (a) de -5 a 4 inclusivé
- (b) de -5 a 5 inclusivé
- (c) de 0 a 5 inclusivé

Considere a seguinte expressão usando a função `rand()` da biblioteca-padrão:

```
int r = rand()%10 + 5;
```

Qual o intervalo de valores que poderá tomar o resultado `r`?

- (a) de 5 a 14 inclusivé
- (b) de 5 a 15 inclusivé
- (c) de 0 a 15 inclusivé

Considere o seguinte fragmento de programa que aplica o algoritmo de Euclides e conta o número de iterações `n` efetuadas.

```
int a=36, b=15, n=0;
while (a != b) {
    if(a > b)
        a = a - b;
    else
        b = b - a;
    n ++;
}
```

Simule passo-a-passo a execução do programa indique quais os valores finais das três variáveis.

- (a) `a=3, b=3, n=5`
- (b) `a=0, b=0, n=3`
- (c) `a=5, b=5, n=3`

Considere o seguinte fragmento de programa que aplica o algoritmo de Euclides e conta o número de iterações n efetuadas.

```
int a=25, b=15, n=0;
while (a != b) {
    if(a > b)
        a = a - b;
    else
        b = b - a;
    n ++;
}
```

Simule passo-a-passo a execução do programa indique quais os valores finais das três variáveis.

- (a) $a=5, b=5, n=3$
- (b) $a=0, b=0, n=3$
- (c) $a=3, b=3, n=5$

Considere a seguinte definição de uma função recursiva:

```
int f(int n) {
    if (n == 0)
        return 1;
    else if(n % 2 == 1)
        return 3 * f(n-1);
    else
        return 2 * f(n-1);
}
```

Qual o resultado calculado por $f(3)$?

- (a) 18
- (b) 12
- (c) 36

Considere a seguinte definição de uma função recursiva:

```
int f(int n) {
    if (n == 0)
        return 1;
```

```

    else if(n % 2 == 0)
        return 3 * f(n-1);
    else
        return 2 * f(n-1);
}

```

Qual o resultado calculado por $f(3)$?

- (a) 18
- (b) 12
- (c) 36

Considere a seguinte função (incompleta) para a pesquisa binária de um valor x numa variável indexada `vec[]` ordenada com tamanho n .

```

int pesquisa_bin(int vec[], int n, int x) {
    int i = 0, j = n-1;
    while (i <= j) {
        int k = i + (j-i)/2; // índice do meio
        if(vec[k] == x)
            return k; // encontrou
        else if (x > vec[k])
            ??????; // instrução A
        else
            ??????; // instrução B
    }
    return -1; // não encontrou
}

```

Como deve completar o programa para implementar corretamente a pesquisa binária?

- (a) instrução A: $i=k+1$; instrução B: $j=k-1$;
- (b) instrução A: $k=i+1$; instrução B: $k=j-1$;
- (c) instrução A: $i=k-1$; instrução B: $j=k+1$;

Considere a seguinte função (incompleta) que copia a cadeia de caracteres em origem para dest

```

void copiar(char *dest, char *origem) {
    while(*origem != '\0')
        ??????; // instrução em falta
}

```



```
    *dest = '\0'; // colocar terminador  
}
```

Como deve completar a instrução em falta?

- (a) `*dest++ = *origem++;`
- (b) `*origem++ = *dest++;`
- (c) `*dest = origem++;`