

Introdução à Programação (CC111)

2011/2012

Exame (21.01.2012)

*duração: 3 horas***Cotação:** 2 + 0.5 + 0.5, 1.5 + 2 + 2 + 1.5, 1.5, 0.5 + 0.5, 1, 3, 0.3 + 0.7 + 1.5 + 1N.º Nome

1. Escreva um programa que leia três inteiros x , m e n , e imprima a tabela das potências x^k de x , para k pertencente ao intervalo $[m, n]$, sabendo-se que $0 \leq m < n \leq 30$ e $|x| \leq 20$. A tabela deve estar (mais ou menos) bem formatada, como no exemplo abaixo, em que $x = -3$, $m = 8$ e $n = 10$.

k	x^k

8	6561
9	-19683
10	59049

Na implementação use variáveis do tipo `int`. Inclua comentários em pontos estratégicos do programa (por exemplo, sobre condições que as variáveis satisfaçam) e que permitam perceber a correção do programa, a menos de erros de *overflow* e *underflow*.

Dê exemplo de um terno x , m e n tal que a tabela obtida tenha erros. Justifique.

(CONTINUA, v.p.f.)

2. Pretende-se um programa completo para analisar uma sequência de n inteiros e apresentar o diagrama das frequências absolutas (i.e., número de ocorrências) dos inteiros do intervalo $[a, b]$ que nela ocorram, como no exemplo dado à direita. Os valores de a , b e n são os três primeiros inteiros lidos.

Exemplo de dados e resultado:

```
-3 4 17 1 -7 1 -2 2 0 0 2 0 2 -3 -3 0 4 5 3 -2

      *
      * *
    ** ***
    ** *****
    -----
```

O programa deve incluir as seguintes funções/declarações:

- a) função principal e declarações relevantes (para intervalos $[a, b]$ com **30** elementos ou menos).
- b) `void conta(int a, int b, int n, int fs[])` que lê da entrada padrão n inteiros, um a um, e regista o número de ocorrências de cada valor do intervalo $[a, b]$. Admita que o vetor `fs` tem espaço suficiente para os contadores necessários, nada mais se sabendo. Não poderá usar nenhum outro vetor na implementação. Como no exemplo, os n inteiros lidos podem incluir valores que não pertencem ao intervalo. (NB: Com penalização de 0.5 valores pode supor que a é 0).
- c) `int maxv(int v[], int nv)` que determina o máximo dos inteiros (positivos, negativos ou zero) guardados nas primeiras nv posições de um vetor dado por `v`, sendo nv maior ou igual a 1.
- d) `void diagr(int fs[], int nfs, char s)` para apresentar o histograma vertical correspondente às nfs frequências dadas. Escreve o histograma por linhas: cada posição terá o caracter dado em `s` (que no exemplo é `*` e indica uma unidade) ou um espaço (se estiver vazia); na última linha tem um caracter `-` por cada valor do intervalo $[a, b]$, por ordem. A função pode alterar o vetor `fs` se for útil.

// continuação da resposta

(CONTINUA, v.p.f.)

3. Escreva uma função `int idade_anos(int dnasc, int dref)` que dada a data de nascimento de um ser vivo e uma data de referência não anterior, retorna o seu número de anos (por exemplo, para 19920315 e 20000107, obtém 7). As datas são inteiros superiores a 19000000 e com oito dígitos.

4. a) Apresente os valores das variáveis `k`, `n` e `t`, e da expressão `t*t`, na execução de `func(42)`, nos três pontos assinalados (em `//1` e `//2` indique-os no momento em que a condição é avaliada).

```
int func(int n) {  
    int k, t;  
    k = 1;  
    for(t=2; t*t < n; t++)    //1  
        if (n%t == 0) k += 2;  
    if (t*t == n) k++;        //2  
    return k;                 //3  
}
```

b) Justifique que se `n` for maior ou igual a 2, o valor de `func(n)` é sempre o número de divisores positivos de `n` que são distintos de `n`.

5. Apresente o *output* do programa seguinte numa execução em que o endereço de **p** é 0xbffc6744, o endereço de **q** é 0xbffc6740, e o vetor fica colocado a partir da posição de endereço 0xbffc6748 (prefixo 0x por o valor ser escrito em hexadecimal). Explique.

```
#include <stdio.h>
int main() {
    int x[3] = {2,-1,-4};
    int *p, *q;

    p = x;
    q = p;

    printf("%d %d %d",x[0],*p,p[1]);
    printf(" %d %d\n",*(p+1),*(p+2));
    printf("%p %p\n",&x[0],&p);
    printf("%p %p\n",p,q);
    return 0;
}
```

6. Escreva uma função `int testa_compara(char pal1[],char pal2[])` para verificar se as duas *strings* dadas por **pal1** e **pal2** são formadas apenas por letras maiúsculas (e o terminador) e se forem, retornar -3 se a primeira aparecer antes da segunda no dicionário, 3 se aparecer depois, e 0 se forem iguais. Se ocorrerem outros caracteres, retorna o valor da constante **INVALIDA** (que definimos como -10000). Na resolução não pode usar funções da biblioteca **string.h**.

7. Para um jardim zoológico, pretende-se um programa para obter a lista de contactos (nome e telefone) dos tratadores dos animais que já viveram mais anos do que a esperança de vida da sua espécie (ver alínea c)). Admita que uma função `void carregaDados()` lerá os dados sobre o jardim zoológico, carregando-os nas seis variáveis globais seguintes.

```
ESPECIE Especie[MAXESPS];
TRATADOR Tratador[MAXTRATS];
ANIMAL Animal[MAXANIMS];
int Ne, Nt, Na;
```

`Ne`, `Nt` e `Na` representam o número de espécies, tratadores e animais existentes no jardim zoológico e não ultrapassam os valores das constantes simbólicas `MAXESPS`, `MAXTRATS` e `MAXANIMS`.

O tipo `ESPECIE` é uma estrutura que permite guardar o nome científico de uma espécie e a esperança média de vida dos animais dessa espécie.

O tipo `ANIMAL` é uma estrutura que permite guardar dados sobre um animal existente: número de registo, identificador da espécie (dado pelo índice da posição na tabela `Especie`), data de nascimento (por exemplo, 20101217, que identifica o ano, mês e dia), nome do país de origem, e número de bilhete de identidade do tratador.

O tipo `TRATADOR` é uma estrutura que permite guardar dados sobre um tratador: número do bilhete de identidade, data de nascimento, telefone, nome e morada.

Todos os nomes e moradas são sequências de caracteres bem formadas e com no máximo `MAXC` caracteres, incluindo o terminador. Todos os restantes valores são **inteiros**.

a) Represente esquematicamente o tipo `ANIMAL` (com indicação dos nomes e tipos dos campos) e o vetor `Animal`.

Implemente o programa, incluindo e usando as seguintes funções/declarações:

b) `int main() { carregaDados(); contactar(20120121); return 0; }`, exatamente, e declarações das variáveis globais, das constantes simbólicas, da função `carregaDados()` com anotação do local onde a definiria (mas não a implemente), de outras funções usadas, dos tipos mencionados, tudo segundo a ordem correta (e arbitrando valores, se necessário), sendo `ESPECIE` definida por:

```
typedef struct {char nome[MAXC]; int vmed;} ESPECIE;
```

c) `int procura_tratador(int bi)` que retorna o índice da posição de `Tratador` que contém os dados do tratador cujo número de um bilhete de identidade é `bi`. O valor de retorno é `-1` se o número de bilhete de identidade não corresponder a nenhum dos existentes. Sabe-se que o vetor `Tratador` está ordenado por ordem crescente de números de bilhete de identidade, pelo que será mais valorizada (0.5 val.) uma implementação baseada em pesquisa binária.

d) `void contactar(int dataref)` para imprimir (na saída padrão) o **nome** e **telefone** do tratador de cada animal que à data de referência `dataref` tenha vivido mais anos do que a esperança de vida da sua espécie. O nome e telefone devem ser escritos numa linha juntamente com o **número de registo** do animal correspondente (assuma não há qualquer erro ou inconsistência nos dados). Como funções auxiliares, deve usar a função definida na alínea anterior e a descrita na questão **3**.

Resolva a questão 7. numa folha de exame (solicite-a p.f.)