Lista de Problemas para o Exercício Programa - MIPS
Organização de Computadores Digitais
1º. semestre de 2018

1. Um número *a* é dito *permutação* de um número *b* se os dígitos de *a* formam uma permutação dos dígitos de *b*.

Exemplo: 5412434 é uma permutação de 4321445, mas não é uma permutação de 4312455.

Obs.: Considere que o dígito 0 (zero) não aparece nos números.

(a) Faça uma função *contadígitos* que dados um inteiro *n* e um inteiro *d*, $0 < d \leq 9$, devolve quantas vezes o dígito *d* aparece em *n*.

(b) Usando a função do item anterior, faça um programa que lê dois inteiros positivos *a* e *b* e responda se *a* é permutação de *b*.

2.a) Construa uma função *encaixa* que dados dois inteiros positivos *a* e *b* verifica se *b* corresponde aos últimos dígitos de *a*.

Ex.:

a	*b*	
567890	890	=> encaixa
1243	1243	=> encaixa
2457	245	=> não encaixa
457	2457	=> não encaixa

(b) Usando a função do item anterior, faça um programa que lê dois inteiros positivos *a* e *b* e verifica se o menor deles é segmento do outro.

Exemplo:

a	*b*	
567890	678	=> b é segmento de a
1243	2212435	=> a é segmento de b

235 236 => um não é segmento do outro

3. Uma seqüência de *n* números inteiros não nulos é dita *piramidal m-alternante* se é constituída por *m* segmentos: o primeiro com um elemento, o segundo com dois elementos e assim por diante até o *m*-ésimo, com *m* elementos. Além disso, os elementos de um mesmo segmento devem ser todos pares ou todos ímpares e para cada segmento, se seus elementos forem todos pares (ímpares), os elementos do segmento seguinte devem ser todos ímpares (pares).

Por exemplo, a seqüência com *n* = 10 elementos:

12 3 7 2 10 4 5 13 5 11 é piramidal 4-alternante.

A seqüência com *n* = 3 elementos:

7 10 2 é piramidal 2-alternante.

A seqüência com *n* = 8 elementos:

1 12 4 3 13 5 12 6 não é piramidal alternante pois o último segmento não tem tamanho 4.

(a) Escreva uma função *bloco* que recebe como parâmetro um inteiro *n* e lê *n* inteiros do teclado, devolvendo um dos seguintes valores:

0, se os *n* números lidos forem pares;
1, se os *n* números lidos forem ímpares;
-1, se entre os *n* números lidos há números com paridades diferentes.

(b) usando a função do item anterior, escreva um programa que, dados um inteiro $n \geq 1$ e uma seqüência de *n* números inteiros, verifica se ela é piramidal *m*-alternante. O programa deve imprimir o valor de *m* ou dar a resposta *não*.

4. (a) Escreva uma função que recebe um inteiro positivo *m* e devolve 1 se *m* é primo, 0 em caso contrário.

(b) Escreva um programa que leia um inteiro não-negativo *n* e imprima a soma dos *n* primeiros números primos.

5. (a) Escreva uma função que recebe como parâmetros dois números *a* e *b* e devolve o *mdc* (máximo divisor comum) de *a* e *b*, calculado por meio do algoritmo de Euclides.

(b) Escreva um programa que leia um inteiro positivo *n* e uma seqüência de *n* inteiros não-negativos e imprime o *mdc* de todos os números da seqüência.

6. (a) Escreva uma função que recebe um número inteiro *n*>0 e devolve o número de dígitos de *n* e o primeiro dígito de *n*.

(b) Escreva um programa que leia uma seqüência de *n* inteiros positivos e imprime o número de dígitos e o primeiro dígito de cada um deles.

7. (a) Escreva uma função que recebe como parâmetro um inteiro positivo *ano* e devolve 1 se *ano* for bissexto, 0 em caso contrário. (Um ano é bissexto se `(ano % 4 == 0 && (ano % 100 != 0 || ano % 400 == 0))`.)

(b) Escreva uma função que tem como parâmetros de entrada e saída três números inteiros, *dia*, *mes* e *ano*, representando uma data, e modifica esses inteiros de forma que eles representem o dia seguinte.

(c) Escreva um programa que leia um inteiro positivo *n* e uma seqüência de *n* datas e imprime, para cada data, o dia seguinte.

8. (a) Escreva uma função de cabeçalho `int divide (int *m, int *n, int d)` que recebe três inteiros positivos como parâmetros e devolve 1 se *d* divide pelo menos um entre **m* e **n*, 0 caso contrário. Fora isso, se *d* divide **m*, divide **m* por *d*, e o mesmo para o **n*.

(b) Escreva um programa que lê dois inteiros positivos *m* e *n* e calcula, usando a função acima, o mínimo múltiplo comum entre *m* e *n*, ou seja, *mmc(m,n)*.

9. (a) Escreva uma função com protótipo `void somabit (int b1, int b2, int *vaium, int *soma);` que recebe três bits (inteiros entre 0 e 1) *b1*, *b2* e **vaium* e devolve um bit soma representando a soma dos três e o novo um bit "vai-um" em *vaium.

(b) Escreva um programa que leia dois números em binário e calcula um número em binário que é a soma dos dois números dados. Utilize a função acima.

10. (a) Escreva uma função com o protótipo `void converte (char ch, int *tipo, char *valor);` que recebe um caractere *ch* e devolve em *tipo 0, se o caractere for um número inteiro, 1 se for uma letra (maiúscula ou minúscula) e 2 caso contrário; e além disso, no caso de ser uma letra, converte para maiúscula, senão devolve ch inalterado.

(b) Escreva um programa que leia uma seqüência de *n* caracteres e imprima a seqüência convertida para maiúscula, eliminando os caracteres que não forem letras ou números.

11. Considere as seguintes fórmulas de recorrências:

$$\begin{cases} F_1 = 2; \\ F_2 = 1; \\ F_i = 2 * F_{i-1} + G_{i-2} \quad i \geq 3 \end{cases} \qquad \begin{cases} G_1 = 1; \\ G_2 = 2; \\ G_i = G_{i-1} + 3 * F_{i-2} \quad i \geq 3 \end{cases}$$

Podemos então montar a seguinte tabela:

i	1	2	3	4	5	...
F_i	2	1	3	8	24	...
G_i	1	2	8	11	20	...

Este exercício está dividido em três partes.

(a) Só para ver se você entendeu as fórmulas, qual é o valor de F_6 e G_6?

(b) Faça uma função de nome *valor* que recebe um inteiro $k \geq 1$ e devolve F_k e G_k.

Exemplo: Para k=2, a função deve devolver os valores 1 e 2. Para k=3, a função deve devolver os valores 3 e 8. Para k=4, a função deve devolver os valores 8 e 11.

(c) Faça um programa que lê um inteiro n > 2 e imprime os valores $F_{n-2} + G_{n+200}$ e $F_{n+200} - G_{n-2}$.

Seu programa deve obrigatoriamente utilizar a função do item anterior, mesmo que você não a tenha feito.

12. Um conjunto pode ser representado por um vetor da seguinte forma: V[0] é o tamanho do conjunto; V[1], V[2], etc. são os elementos do conjunto (sem repetições).

(a) Faça uma **função** chamada *intersecção* que dados dois conjuntos de números inteiros *A* e *B*, constrói um terceiro conjunto *C* que é a intersecção de *A* e *B*. Lembre-se de que em *C*[0] a sua função deve colocar o tamanho da intersecção.

(b) Faça um **programa** que lê um inteiro $n \geq 1$ e uma seqüência de *n* conjuntos de números inteiros (cada um com no máximo 100 elementos) e constrói e imprime um vetor INTER que representa a intersecção dos *n* conjuntos.

Por exemplo, se *n*=3 e os conjuntos são {1, 2, 4, 9}, {2, 4, 7, 8, 9} e {5, 4, 9}, a entrada será:

3	O valor de *n*
4	V[0] = tamanho do primeiro conjunto
1 2 4 9	V[1] V[2] V[3] V[4]
5	V[0] = tamanho do segundo conjunto
2 4 7 8 9	V[1] V[2] V[3] V[4] V[5]
3	V[0] = tamanho do terceiro conjunto
5 4 9	V[1] V[2] V[3]

E o vetor INTER construído será

INTER[0] = 2		tamanho do conjunto
INTER[1] = 4	INTER[2] = 9	conjunto intersecção

NOTE que não é preciso ler todos os conjuntos de uma só vez. Você pode ler os dois primeiros conjuntos e calcular a primeira intersecção. Depois, leia o próximo conjunto e calcule uma nova intesecção entre esse conjunto lido e o conjunto da intersecção anterior, e assim por diante.

Use obrigatoriamente a função do item anterior, mesmo que você não a tenha feito.

13. (a) Escreva uma função que lê, linha a linha, uma matriz real A_{mxn}

(b) Escreva uma função que imprime uma matriz real A_{mxn}

14. (a) Escreva uma função que calcula a soma dos elementos da linha *i* de uma matriz real A_{mxn}.

(b) Escreva uma função que calcula o produto dos elementos da coluna *j* de uma matriz real A_{mxn}.

15. (a) Escreva uma função que troca o conteúdo de duas variáveis.

(b) Escreva uma função que recebe dois inteiros, *i* e *j*, uma matriz real A_{mxn} e troca linha *i* pela linha *j*. Utilize a função do item anterior.

16.

(a) Faça uma função *MAX* que recebe como entrada um inteiro *n*, uma matriz inteira A_{nxn} e devolve três inteiros: *k*, *Lin* e *Col*. O inteiro *k* é um maior elemento de *A* e é igual a *A*[*Lin*,*Col*].

Exemplo:

$$se \quad A = \begin{pmatrix} 3 & 7 & 1 \\ 1 & 2 & 8 \\ 5 & 3 & 4 \end{pmatrix} \quad então \quad \begin{cases} k = 8 \\ Lin = 1 \\ Col = 2 \end{cases}$$

Obs.: Se o elemento máximo ocorrer mais de uma vez, indique em *Lin* e *Col* qualquer uma das possíveis posições.

(b) Faça um programa que, dado um inteiro *n* e uma matriz quadrada de ordem *n*, cujos elementos são todos inteiros positivos, imprime uma tabela onde os elementos são listados em ordem decrescente, acompanhados da indicação de linha e coluna a que pertencem. Havendo repetições de elementos na matriz, a ordem é irrelevante. Utilize obrigatoriamente o procedimento da parte (a), mesmo que você não o tenha feito.

Ex.: No caso da matriz acima, a saída poderia ser:

Elem	Linha	Coluna
8	1	2
7	0	1
5	2	0
4	2	2
3	0	0
3	2	1
2	1	1
1	0	2
1	1	0

17. Escreva uma função que recebe uma matriz de caracteres 8x8 representando um tabuleiro de xadrez e calcula o valor total das peças do jogo. Espaços vazios do tabuleiro são codificados como casas com ` '(branco) e têm valor 0 (zero). O valor das demais peças é dado de acordo com a tabela:

Peça	Valor
peão	1
cavalo	3
bispo	3
torre	5
rainha	10
rei	50

18. (a) Escreva uma função que recebe como parâmetros um vetor real *A* com *n* elementos e um vetor real *B* com *m* elementos, ambos representando conjuntos, e verifica se *A* está contido em *B* ($A \subseteq B$).

(b) Usando a função do item acima verifique se dois conjuntos são iguais ($A = B$ *se e somente se* $A \subseteq B$ *e* $B \subseteq A$) .

19. (a) Escreva uma função que recebe uma matriz real A_{mxn} e determina a sua transposta (Se *B* é a matriz *transposta* de *A* então $a_{ij} = b_{ji}$).

(b) Escreva uma função que calcula o produto escalar de dois vetores dados.

(c) Dada uma matriz real X_{mxn}, usando as funções acimas, calcule o produto X x X^t .

(d) Faça uma função que verifica se uma matriz A_{mxm} é a matriz identidade.

(e) Ache uma aplicação para esse exercício.

20. (a) Dado um vetor real *X* com *n* elementos e um certo índice *k*, escreva uma função que determina o índice do elemento mínimo entre X[k], X[k+1], ..., X[n-1].

(b) Usando a função do item anterior coloque os elementos de um vetor em ordem crescente.

21. Para encontrar uma raiz de um polinômio $p(x)=a_0+a_1x+...+a_nx^n$, ($n \geq 2$), pode-se aplicar o método de Newton (1), que consiste em refinar uma aproximação inicial x_0 dessa raiz através da expressão:

$$x_{n+1} = x_n - \frac{p(x_n)}{p'(x_n)}$$

n= 0,1,2,..., onde $p'(x)$ é a primeira derivada de $p(x)$.

Usualmente, repete-se esse refinamento até que $|x_{n+1} - x_n| < \varepsilon$, $\varepsilon \geq 0$, ou até que *m* iterações tenham sido executadas.

Dados um polinômio $p(x)=a_0+a_1x+...+a_nx^n$, uma aproximação inicial x_0 da raiz de $p(x)$, $\varepsilon \geq 0$ e o número máximo de iterações que devem ser executadas, determine uma aproximação da raiz de $p(x)$ pelo método de Newton. Utilize uma função que, dado um polinômio $p(x)$, calcula a derivada $p'(x)$ e, para calcular $p(x_n)$ e $p'(x_n)$ em cada iteração, uma função que calcula o valor de um polinômio em um ponto.

22. (a) Escreva uma função que recebe como parâmetros:

- dois inteiros positivos *n* e *m*;
- uma matriz A_{nxm} ;
- o índice *c* de uma coluna;
- os índices *k* e *p* de duas linhas,

e ordena entre as linhas *k* e *p* da matriz *A* segundo a coluna *c*.

Exemplo:
n=6, m=3
c=1,
k=0, p=4

$$A = \begin{pmatrix} 0 & 12 & 1 \\ 1 & 2 & -1 \\ 0 & -2 & 3 \\ -1 & -3 & 2 \\ 0 & 18 & 1 \\ 0 & 10 & 1 \end{pmatrix} \quad \textit{tem} \quad \textit{como} \quad \textit{saída} \quad \begin{pmatrix} -1 & -3 & 2 \\ 0 & -2 & 3 \\ 1 & 2 & -1 \\ 0 & 12 & 1 \\ 0 & 18 & 1 \\ 0 & 10 & 1 \end{pmatrix}$$

(b) Dadas *n* datas em uma matriz $DATA_{nx3}$, onde a primeira coluna corresponde ao dia, a segunda ao mês e a terceira ao ano, coloque essas datas em ordem cronológica crescente, usando a função acima.

Exemplo: $n = 6$,

$$DATA = \begin{pmatrix} 25 & 6 & 1965 \\ 16 & 6 & 1965 \\ 13 & 12 & 1941 \\ 21 & 4 & 1965 \\ 6 & 2 & 1989 \\ 1 & 10 & 1973 \end{pmatrix} \quad tem \quad como \quad saída \quad \begin{pmatrix} 13 & 12 & 1941 \\ 21 & 4 & 1965 \\ 16 & 6 & 1965 \\ 25 & 6 & 1965 \\ 1 & 10 & 1973 \\ 6 & 2 & 1989 \end{pmatrix}$$

23. (a) Escreva uma função que recebe como parâmetros uma matriz de caracteres $NOMES_{mxn}$, os índices *i* e *j* de duas linhas e que troca os elementos da linha *i* com os da linha *j*.

(b) Escreva uma função que recebe como parâmetros uma matriz $NOMES_{mxn}$, os índices *i* e *j* de duas linhas e que devolve valor 1 se o nome na linha *i* é maior ou igual ao da linha *j* (ordem alfabética) e 0 caso contrário.

(c) São dados *n* nomes que serão armazenados em uma matriz $NOMES_{mxn}$. Coloque os nomes dessa matriz em ordem alfabética usando as funções descritas acima.

24.

(a) Escreva uma função que recebe como parâmetros uma matriz real A_{nxm} , e uma posição (*i*,*j*) da matriz, e calcula a média aritmética dos vizinhos de (*i*,*j*), ou seja, a média entre $A_{i-1,\, j}$, $A_{i+1,\, j}$, $A_{i,\, j+1}$,$A_{i,\, j+1.}$ Desconsidere os vizinhos que não pertencem a matriz (por exemplo, os vizinhos de (0, 0) são somente (0,1) e (1,0)).

(b) Escreva uma função que recebe como parâmetro uma matriz real A_{nxm} e devolve uma matriz $A^{média}$, onde $a_{ij}^{média}$ é a média aritmética dos vizinhos de (*i*,*j*). Para isto, utilize a função do item anterior.

(c) Escreva um programa que lê uma matriz real A_{nxm} , e um número inteiro *k*; utilizando a função do item anterior, o programa deve transformar a matriz *k* vezes, imprimindo a matriz inicial e depois de cada transformação.

25. Dizemos que uma matriz A_{nxn} é um *quadrado latino* de ordem *n* se em cada linha e em cada coluna aparecem todos os inteiros 1,2,3,...,n (ou seja, cada linha e coluna é permutação dos inteiros 1,2,...,n).

Exemplo:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 1 & 2 & 3 \\ 3 & 4 & 1 & 2 \end{pmatrix}$$

A matriz acima é um *quadrado latino* de ordem 4.

(a) Escreva uma função que recebe como parâmetros um inteiro *n*, um vetor *V* com *n* inteiros e verifica se em *V* ocorrem todos os inteiros de 1 a *n*.

(b) Usando a função acima, verifique se uma dada matriz inteira A_{nxn} é um quadrado latino de ordem *n*.

26.

(a) Faça uma função que transforma um número num vetor correspondente à sua representação binária.

(b) Dados inteiros positivos *n* e *m*, e os vetores *ReprN* e *ReprM* correspondentes à representação binária dos números *n* e *m*, considere a seguinte matriz *A* de caracteres:

$$a_{ij} = \begin{cases} \text{`*' se } ReprN[i]= 1 \text{ e } ReprM[j]= 1 \\ \text{` ' caso contrário} \end{cases}$$

Faça um programa que recebe *n* e *m*, e constrói a matriz *A* descrita acima, usando o item (a).

27. Dada uma matriz real quadrada *A* de ordem *n* e um inteiro positivo *k*, define-se a aproximação da matriz real e^A pela soma abaixo:

$$e^A = I_n + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots + \frac{A^k}{k!}$$

sendo I_n a matriz identidade de ordem *n*.

(a) Faça uma função que recebe como parâmetros um inteiro *n* e duas matrizes quadradas reais *X* e *Y* de ordem *n*. Esta função devolve em uma matriz *Z*, também passada como parâmetro, a soma das matrizes *X* e *Y*.

(b) Escreva uma função que recebe como parâmetro um número inteiro *n*, um número real *c* e uma matriz X_{nxn}. A função devolve em uma matriz *Y*, também passada como parâmetro, o produto do número *c* pela matriz *X*. Ou seja,

$$Y_{i,j} = c * X_{i,j} \quad \text{para} \quad 0 \leq i \leq n-1 \quad \text{e} \quad 0 \leq j \leq n-1$$

(c) Escreva uma função que recebe como parâmetros um inteiro *n* e duas matrizes quadradas reais X_{nxn} e Y_{nxn}. Esta função devolve em uma matriz *Z*, também passada como parâmetro, o produto das matrizes *X* e *Y*.

(d) Faça um programa que, dados dois inteiros *n* e *k* e uma matriz real quadrada A_{nxn}, determina uma aproximação da matriz real e^A utilizando **obrigatoriamente** as três funções mencionadas anteriormente.

28. Uma função matemática pode ser representada por um vetor. Por exemplo, com *n* = 5 e o vetor de tamanho *n*[0.0, 0.5, 1.0, 1.5, 2.0] estamos representando a função *f*(*i*)=*i*/2, *i*=0,1,2,3,4.

O alisamento de uma função é definido como:

- $g(i) = \frac{f(i-1) + f(i) + f(i+1)}{3}, \quad para \quad 1 \leq i \leq n-2;$
- $g(0) = g(1)$;
- $g(n\text{-}1) = g(n\text{-}2)$.

Para o exemplo acima, temos:

0.0	0.5	1.0	1.5	2.0	função f
0.5	0.5	1.0	1.5	1.5	alisamento g
0.66...	0.66...	1.0	1.33...	1.33...	alisamento de g

Obs.: Não utilize variáveis globais para escrever as funções abaixo.

(a) Escreva uma função **alisa** que recebe um vetor real F com *n* elementos e devolve um vetor G contendo o alisamento da função representada em F.

(b) Escreva uma função **realisa** que recebe *m*,*n* inteiros e um vetor F de *n* números reais e retorne em uma matriz de números reais A_{mxn} os *m* alisamentos sucessivos da função representada em *F.* Cada vetor deverá ser armazenado em um linha da matriz.

(c) Escreva uma função **avalia** que recebe os números inteiros *m*,*n* e um vetor F de n números reais e, utilizando obrigatoriamente a função do item anterior (se não o fez, escreva apenas o protótipo) retorne quais são os valores máximo e mínimo da representação do m-ésimo alisamento de F.