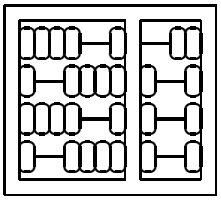


MC404 Organização de Computadores e Linguagem de Montagem

IC – UNICAMP

Segunda Lista de Exercícios

1. Em muitas situações, quando trabalhamos com valores aritméticos, devemos arredondar a resposta usando um certo número de casas decimais. Para arredondar um número decimal somamos 5 à casa decimal a partir da qual desejamos ignorar e depois as ignoramos. Por exemplo: seja o número 429.867 com 3 dígitos decimais e queremos arredonda-lo de forma a usarmos somente dois dígitos decimais após o ponto decimal, assim podemos somar 5 à terceira casa decimal obtendo 429.872, e então ignorar os dígitos a partir da terceira casa decimal obtendo 429.87. Escreva e teste uma subrotina em linguagem de montagem do 8086 que arredonda um valor dado em BCD para dois dígitos decimais após o ponto decimal e imprime o resultado na tela. O valor BCD original é apontado por SI e armazenado com os dígitos mais significativos nos endereços menores e o número de dígitos BCD é dado em BX e o número de dígitos decimais após o ponto decimal é dado em CX.

2. Escreva uma subrotina que é uma generalização da rotina do exercício anterior que tem como entrada mais um parâmetro especificado em BP que fornece o número de dígitos decimais após o ponto decimal que devem ser usados para o arredondamento.

3. Refaça os exercícios anteriores com os parâmetros passados na pilha e não em registradores.

4. Escreva um programa que leia um número inteiro N entre 0 e 9 imprima todas as permutações das N primeiras letras minúsculas do alfabeto. As permutações devem ser geradas em ordem alfabética.

   Exemplo: se N = 3, teremos como saída do programa:

   abc acb bac bca cab cba

   Sugestão: escreva primeiro o programa em alto nível, usando recursão. A seguir traduza para linguagem de montagem manualmente.

5. Escreva duas rotinas VABS e SMC2, descritas abaixo. Além de AX somente os flags podem ser alterados. Não é permitido o uso de pilha (exceto para retorno) ou de posições auxiliares de memória.

   (a) VABS – Recebe em AX um inteiro com sinal na notação complemento de um e retorna em AX o valor absoluto desse inteiro.

   (b) SMC2 – Recebe em AX um inteiro com sinal na notação sinal e magnetude e retorna em AX o mesmo inteiro na notação complemento de dois.

6. Escreva um subrotina que inverta as posições dos valores (bytes) de um vetor sem copia-lo para outra área. Um apontador para o início do vetor e o número de elementos são passados como parâmentros na pilha.

   Exemplo:

   Vetor antes: 10 30 5 15 2 4 20

   Vetor depois: 20 4 2 15 5 30 10

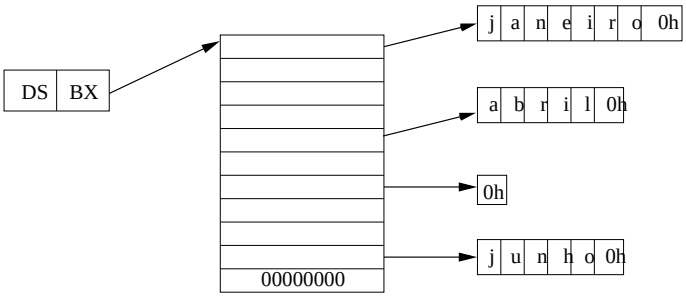
7. Explique como as instruções Call e Ret fazem uso da pilha. Se não houvesse pilha, como você faria para implementar estas instruções (Call e Ret).

8. Escreva um procedimento que verifique se duas palavras de 16 bits passadas como parâmetros na pilha têm o mesmo número de bits 1. Caso ambas tenham o mesmo número, o procedimento deve retornar o bit de estado `CF` desligado e o registrador `AX` deve conter o número de bits 1. Caso contrário, o bit de estado `CF` deve retornar ligado.

9. Escreva um procedimento que verifique se uma cadeia de bytes é palíndrome (ou seja, dá o mesmo resultado quando lida da direita para a esquerda ou da esquerda para a direita; "miauaim´´ é um exemplo de palíndrome). O endereço do início da cadeia é dado na pilha na forma SEGMENTO:OFFSET e o seu comprimento (maior ou igual a zero) também é fornecido na pilha. Se a cadeia é palíndrome, o registrador `Al` deve retornar com zero; caso contrário, `Al` deve retornar com 1.

10. Escreva um procedimento *Compr* que receba como parâmetro (passado na pilha) um apontador para uma cadeia de caracteres e devolva em `AL` o comprimento da cadeia (ou seja, o número de caracteres). A cadeia, possivelmente vazia, é terminada pelo valor 00H (este caracter de terminação não entra no cômputo do comprimento da cadeia).

11. Considere que *TabCadeia* é um vetor onde cada elemento é um apontador (`segmento:offset`) para uma cadeia de caracteres terminada pelo valor 00H, como no exercício anterior. O final do vetor *TabCadeia* é indicado pelo valor 0000H:0000H. Escreva um procedimento *Busca* que recebe em DS:BX o endereço inicial da tabela *TabCadeia* e, utilizando o procedimento *Compr* do exercício anterior, devolva em AL o comprimento da cadeia com o maior número de caracteres e em ES:SI o endereço de início dessa cadeia. Se houver mais de uma cadeia com comprimento máximo deve ser retornado o endereço da primeira cadeia.



0

Figura 1: *TabCadeia*

12. Escreva um procedimento que calcule a soma de dois números inteiros positivos, onde cada inteiro é representado por um vetor de elementos de 4 bits cada (cada elemento representa um dígito do número). O primeiro elemento do vetor indica quantos dígitos tem o número (cada número portanto pode ter até 16 dígitos); o segundo elemento é o dígito *menos significativo*, o último elemento é o dígito *mais significativo*. São passados como argumento, pela pilha, os endereços iniciais dos dois vetores que se deseja somar, e o endereço inicial do vetor resultado.

13. Idem ao exercício anterior porém calculando a diferença entre os dois vetores.