

Software Básico - Turma B 2  
Prova 2

1. (3.5 pontos) O programa em C abaixo chama uma função em Assembly I A-32. A função faz a subtração elemento por elemento de duas matrizes (a matrix resultante é colocada na mesma estrutura de dados que a primeira matriz passada como argumento, por passagem por referência), e retorna um. Faça a função em Assembly que seja compatível com este código, mantenha tipagem, use os endereçamentos adequados para as estruturas de dados, variáveis locais devem ir na pilha e a passagem de parâmetros deve ser feita pela pilha retorno pelo EAX (pode usar IO.MAC).

```
include <stdio.h>
int main( )
{
    int m , n , c , d , matrix[100] , matrix2[100],soma;
    printf( " Enter the number of rows and columns of the matrices (lower than 100)\n" );
    scanf( " %d%d", & m , & n );
    printf( " Enter elements of the matrix 1\n" );
    extern int f1(int * , int);
    for ( c = 0 ; c < m ; c ++ )
        for ( d = 0 ; d < n ; d ++ )
            scanf( " %d", ( matrix+ c*sizeof(int)+d));
    printf( " Enter elements of the matrix 2\n" );
    extern int f1(int*, int*, int, int);
    for ( c = 0 ; c < m ; c ++ )
        for ( d = 0 ; d < n ; d ++ )
            scanf( " %d", ( matrix2+ c*sizeof(int)+d));

    f1(matrix,matrix2,m,n);
    for ( c = 0 ; c < n ; c ++ ) {
        for ( d = 0 ; d < m ; d ++ )
            printf( " %d\t", ( matrix+ c*sizeof(int)+d));
        printf( " \n" );
    } return 0 ;
}
```

2. (3.5 pontos) Faça um programa em Assembly I A-32 que (NAO pode usar IO-MAC):

- Abre um arquivo em modo leitura (00), com permissão para que todos os usuários possam ler, escrever e executar. Nome do arquivo: ' myfile1.txt' (na mesma pasta que o programa). Lê 100 valores de 16 bits do arquivo e os salva num array x .
- Fecha o arquivo.

- Verifica por meio de um laço se os valores lidos são maiores que ZERO. Gerando um array y com valor 1 caso positivo ou 0 caso negativo para cada valor.
- Abre um arquivo em modo escrita (01), com permissão para o dono do arquivo possa ler, escrever e executar e os outros possam somente ler. Nome do arquivo: ' myfile2.txt' (na mesma pasta que o programa). E escreve o array y .
- Fecha o arquivo.

3. (3 pts) Considere um número em ponto flutuante baseado no formato da IEEE. O número é formado por 7 bits. Um bit para o sinal, os próximos quatro bits para o expoente, e os últimos dois bits para a mantissa. Como visto em sala de aula o formato IEEE, possui números normalizados, não normalizados, duas representações de zero, infinito e NaN. Assumindo que arredondamentos são feitos utilizando arredondamento para menos infinito. Preencha a tabela abaixo, na mantissa, expoente e valor deve ser colocado o valor em decimal do código binário de cada campo, no campo binário deve ser colocado o código em binário do número completo. Pode utilizar notação exponencial ( $2^{512}$ ) ou fracionária ( $2 \frac{1}{3}$ ). Quando indica-se o maior/menor número NÃO deve ser considerado infinito

Descrição	Binário	Mantissa	Expoente	Valor em Decimal
Menos Zero	1000000	0	-6.0	-0.0
Número Positivo mais próximo a zero				
Maior Número Normalizado				
Menor Número Não-normalizado				
7.0 + 8.0				