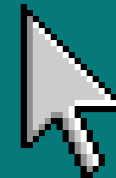


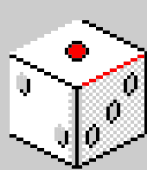
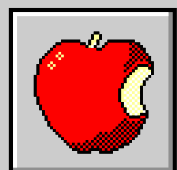
Arquitetura MIPS

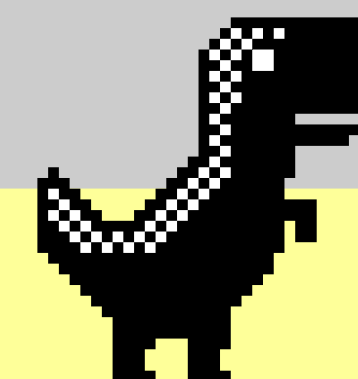
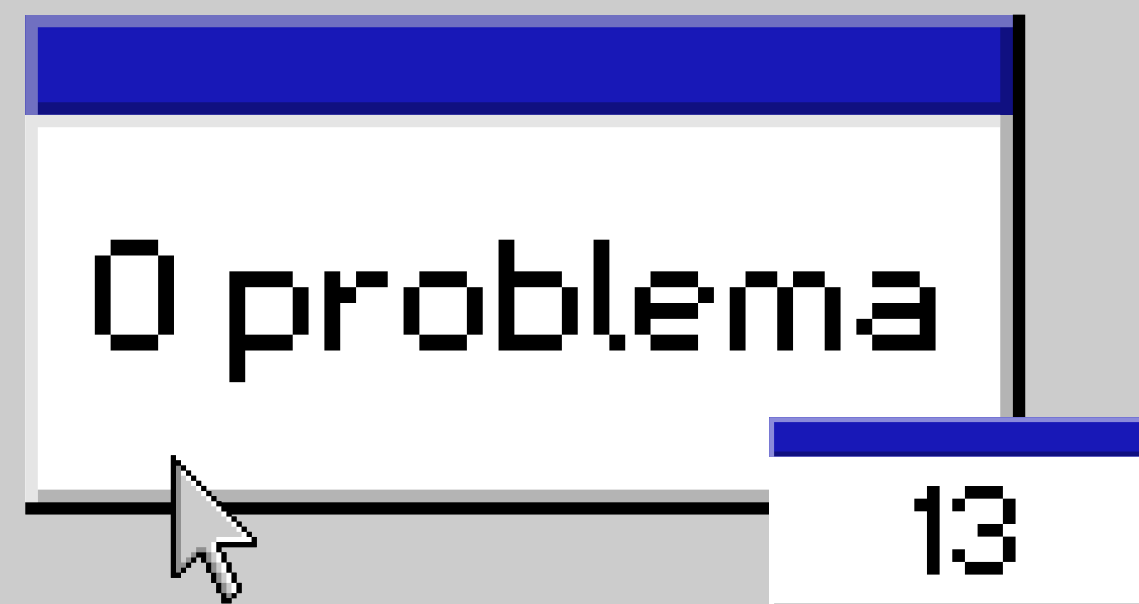


Organização e Arquitetura
de Computadores

Bruno Friedrich
Raquel - 13672450

Matheus Silva Lopes da
Costa - 12674680



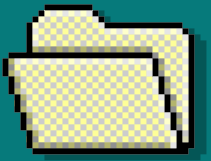


- (a) Escreva uma função que lê, linha a linha, uma matriz real $A_{m \times n}$
- (b) Escreva uma função que imprime uma matriz real $A_{m \times n}$

Deve-se, então, alocar memória suficiente para suportar uma matriz de valores reais, chamada de A .

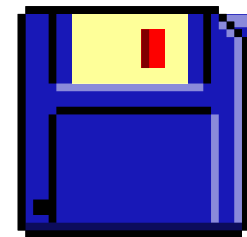
Essa matriz tem dimensão $m \times n$, ou seja, 'm' linhas e 'n' colunas. Além disso, após ter a matriz criada é necessário preenchê-la com a inserção de valores em cada posição e imprimí-los.

Solução



Leitura e
Armazenamento

Do número de
linhas e colunas



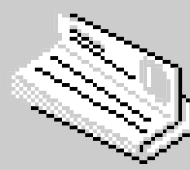
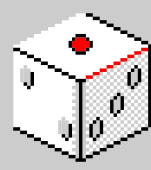
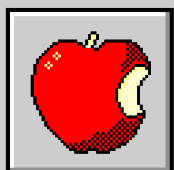
Leitura e
Armazenamento

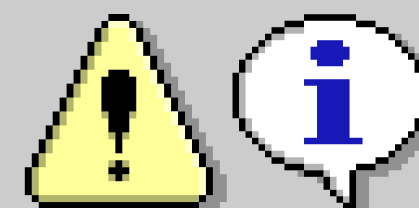
Dos valores a ser
inseridos na matriz



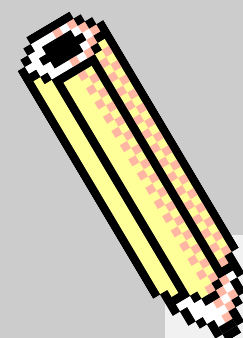
Impressão

Dos valores da
matriz





DADOS ARMAZENADOS NA MEMÓRIA



```

1  #programa que recebe o numero de linhas, o numero de colunas
2  # e uma matriz de float em seguida imprime a matriz
3
4  .data
5      initialProgramText: .ascii "Programa que lê e imprime uma matriz real"
6      getRowsNumberText: .ascii "Insira o numero de linhas: "
7      getColsNumberText: .ascii "Insira o numero de colunas: "
8      insertMatrixText: .ascii "Insira a matriz:"
9      newlineChar: .ascii "\n"
10     spaceChar: .byte ' '
    
```

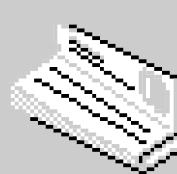
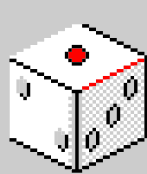
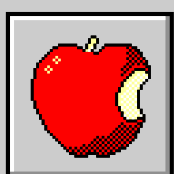
Start

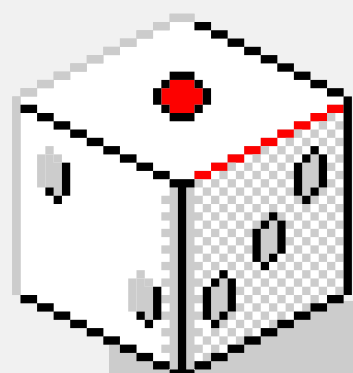
LEITURA DA QUANTIDADE DE LINHAS E COLUNAS

```
11 .text
12     main:
13         la $a0,initialProgramText #carrega o endereço de initialProgramText em $a0
14         li $v0,4 #carrega o valor 4 em $v0
15         syscall # chamada do sistema para imprimir initialProgramText
16
17         jal newLine # chama subrotina para imprimir quebra de linha
18
19         la $a0,getRowsNumberText #carrega o endereço de getRowsNumberText em $a0
20         li $v0,4 #carrega o valor 4 em $v0 código de impressão de string
21         syscall # chamada do sistema para imprimir getRowsNumberText
22
23         jal newLine
24
25         li $v0,5 #carrega o valor 5 em $v0 código de leitura inteiro
26         syscall # chamada do sistema para ler um int e armazenar em $v0
27
28         add $s0,$v0,$zero #salvando numero de linhas
29
30         la $a0,getColsNumberText #carrega o endereço de getRowsNumberText em $a0
31         li $v0,4 #carrega o valor 4 em $v0 código de impressão de string
32         syscall # chamada do sistema para imprimir getRowsNumberText
```

CÁLCULO DA QUANTIDADE DE FLOATS E BYTES

```
35     jal newLine # chama subrotina para imprimir quebra de linha
36
37     li $v0,5 #carrega o valor 5 em $v0 código de leitura inteiro
38     syscall # chamada do sistema para ler um int e armazenar em $v0
39
40     add $s1,$v0,$zero #salvando o numero de colunas
41
42     mul $t0,$s0,$s1 #calculando numero de floats que serao recebidos
43
44     mul $t1,$t0,4 #calculando numero de bytes para armazenamento e salvando em $t3
45
46     add $a0,$zero,$t1 #carregando em $a0 o numero de bytes necessarios
47     li $v0,9 # codigo para solicitar reserva de memória heap
48     syscall # chamada do sistema para alocar memoria
49
50     div $t1,$s0 #calculando tamanho em bytes de uma linha
51     mflo $s3 #salvando tamanho em bytes de uma linha
52
53     add $s2,$zero,$v0 # salvando o endereco inicial da sequencia de bytes
54
55     add $t1,$zero,$t0 # iniciando contador de elementos
56
57     add $t0,$zero,$v0 # iniciando o iterador para armazenagem na pilha
```





LEITURA DOS FLOATS

```

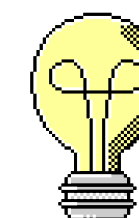
59      la $a0,insertMatrixText #carrega o endereço de getRowsNumberText em $a0
60      li $v0,4 #carrega o valor 4 em $v0 código de impressão de string
61      syscall # chamada do sistema para imprimir getRowsNumberText
62
63      jal newLine
64
65      readLoop:
66          li $v0,6 # carregando em $v0 código de leitura de float
67          syscall # chamada do sistema para ler um float
68          s.s $f0,($t0) # armazenando o valor lido na "matriz"
69          addiu $t0,$t0,4 # apontando para o espaço de armazenamento seguinte
70          subi $t1,$t1,1 # decrementa o contador de elementos
71          bne $t1,$zero,readLoop # carrega o endereço de readLoop em $pc se $t1 for diferente de $zero
72
73      jal newLine
74
75      add $t2,$zero,$s2 # iniciando iterador de impressão
    
```



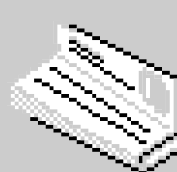
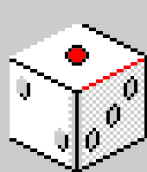
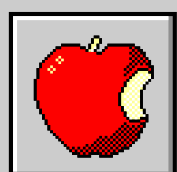

```

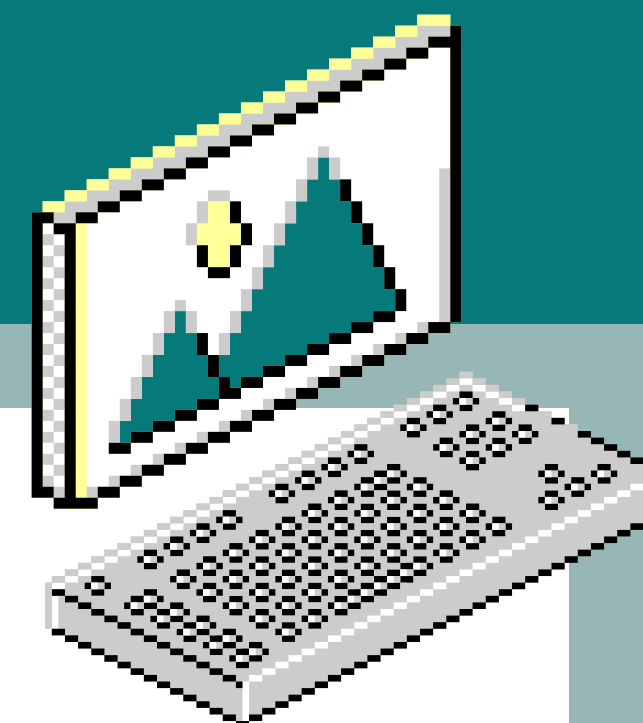
77     printLineLoop:
78     beq $t2,$t0,exit_sucess
79     add $sp,$sp,-4
80     sw $t2,($sp)
81     jal imprimeLinha
82     jal newLine
83     add $t2,$t2,$s3
84     add $sp,$sp,4
85     j printLineLoop
86 imprimeLinha:
87     lw $t1,($sp) # armazena o valor de ($sp) em $t1 para imprimir o valor
88     add $t3,$t1,$s3
89     printColLoop:
90         beq $t3,$t1,return
91         l.s $f12,($t1)
92         li $v0,2 # armazena em $v0 codigo para impressão de float
93         syscall # chamada do sistema para impressão de float
94
95         la $a0,spaceChar # carrega o endereço de spaceChar é armazenado em $a0
96         li $v0,4 # armazena o codigo de impressao de char em $v0
97         syscall # chamada do sistema para impressao de espaco
98         addiu $t1,$t1,4
99     j printColLoop

```



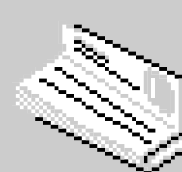
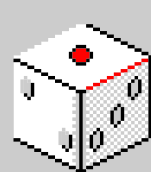
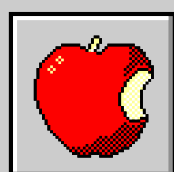
IMPRIME , LINHA A
LINHA, A MATRIZ



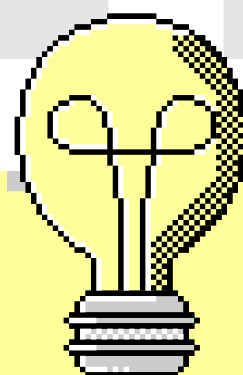


```
100      return:
101      jr $ra # carrega em $pc o valor de $ra
102
103      newLine:
104          la $a0, newLineChar #Carrega o endereço de newLine em $a0
105          li $v0, 4 # carrega o código de impressão de caractere em $v0
106          syscall # chamada do sistema para imprimir \n
107          jr $ra
108
109      exit_sucess:
110          li $v0, 10
111          syscall
```

ROTINA PARA QUEBRA DE LINHA E
ENCERRAR PROGRAMA



Exemplo



Passo 1: Cria uma matriz com 3 linhas e 4 colunas

3
4

Passo 2: Digite todos os 12 valores

1.1	2.7	3.2	4.2
3.8	9.3	7.2	6.5
1.7	9.6	7.2	8.3

```

Saída:

1.1 2.7 3.2 4.2
3.8 9.3 7.2 6.5
1.7 9.6 7.2 8.3

-- program is finished running --
    
```





Obrigado!

Bruno Friedrich Raquel - 13672450

Matheus Silva Lopes da Costa - 12674680