

## AV2020QUNICA:

Construir um código iterativo que calcule o n-éssimo número fibonacci.

fib(n)

### Requisitos:

- Código montável, ligável e executável
- O cálculo do n-éssimo número fibonacci deve ser iterativo
  - não usar recursividade, pois não estudamos funções até o momento
- Sintaxe Intel x64\_86

### Funcionamento:

- Deve ser solicitado ao usuário a entrada do n-éssimo número fibonacci buscado
  - A entrada é pelo teclado
    - uma string de caracteres ASCII que representa o número
    - máximo com 2 dígitos
      - usuário entrará sempre com 0, 1, 2 ou +3 dígitos e enter para finalizar
      - nunca caracteres alfabéticos ou especiais.
- Verificação de entrada
  - 1 ou 2 dígitos
    - verificação de limites de representação
      - Quanto é fib(99)?
      - Qual é o tamanho do registrador x64\_86?
  - n=0 ou +3 dígitos
    - mensagem de falha genérica, limpeza de buffer e encerramento
- Conversão dos dígitos ASCII para número equivalente
  - dicas:
    - ASCII para todos os números são:

ASCII:	0011 0000	0011 0001	0011 0010	0011 0011	0011 0100	0011 0101	0011 0110	0011 0111	0011 0100	0011 0101
HEX:	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x39
Dígito:	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
Num:	0	1	2	3	4	5	6	7	8	9

- Após, usar a notação posicional
    - Exemplo1:  $5_d$  é escrito como  $(0) + 5$
    - Exemplo2:  $34_d$  é escrito como  $(10*3) + 4$
    - Exemplo3:  $57_d$  é escrito como  $(10*5) + 7$
  - Cuidado: verifique no gdb a ordem dos dígitos na memória!
- Caso a conversão tenha sucesso e o limite não tenha sido excedido
  - Calcular, de forma iterativa, fibonacci
    - fib(0) = 0
    - fib(1) = 1
    - fib(2) = fib(1) + fib(0)
    - fib(i) = fib(i-1) + fib(i-2)
  - Utilize os links para verificar sua solução.
    - [WolframAlpha](#) ou [Ke!sanOC](#)
- Para finalizar, grave um arquivo binário
  - Nome do arquivo: fib(n).bin, onde n é a entrada do usuário
  - Conteúdo: resultado em "formato" binário
    - isto é, não é necessário converter inteiro para caracteres ASCII.