UTILIZANDO NOSSA MÁQUINA HIPOTÉTICA VAMOS CONSTRUIR UM PROGRAMA PARA CONVERTER VALORES DE UMA UNIDADE PARA OUTRA. O NOSSO PROGRAMA RECEBE UM VALOR NUMÉRICO QUE CORRESPONDE A UMA TEMPERATURA EM GRAUS CELSIUS E EXIBE A TEMPERATURA CORRESPONDENTE EM GRAUS FAHRENHEIT.

A FÓRMULA DE CONVERSÃO ENTRE AS UNIDADES É: $F = 1.8\,C + 32.0$

PROGRAMA:

```
READ 0
STORECONST 1.8 1
MUL 1 0
STORE 2
STORECONST 32 3
ADD 2 3
STORE 4
WRITE 4
```

0	30.
1	1.8
2	54.
3	32.
4	86.

86.

COM AS INSTRUÇÕES VISTA ATÉ AQUI A NOSSA MÁQUINA HIPOTÉTICA É SEQUENCIAL E É CAPAZ DE IMPLEMENTAR APENAS PROGRAMAS QUE APLICAM DIRETAMENTE FÓRMULAS MATEMÁTICAS (FORTRAN John Warner Backus 1954).

MUITOS PROBLEMAS MAIS COMPLEXOS PODEM SER RESOLVIDOS DE FORMA INCREMENTAL: REALIZANDO CÁLCULOS SIMPLES, REPETIDAMENTE.

O QUE ESTARIA FALTANDO PARA MELHORAR A NOSSA MÁQUINA?

COMO FAZER NOSSA MÁQUINA EXECUTAR REPETIDAMENTE UM TRECHO DE INSTRUÇÕES?

ADICIONANDO A ELA UMA NOVA INSTRUÇÃO:

JUMP pos offset

SALTE PARA A INSTRUÇÃO X SE O CONTEÚDO DE pos FOR MAIOR QUE ZERO.

ENDEREÇO DE X = ENDEREÇO ATUAL + offset

O QUE FAZ O PROGRAMA ABAIXO?

```
READ 0
STORECONST 1 1
WRITE 0
SUB 0 1
STORE 0
JUMP 0 -3
```

VAMOS AGORA CONSTRUIR UM PROGRAMA QUE FAÇA NOSSA MAQUININHA CALCULAR O FATORIAL DE UM NUMÉRO:

$$N! = \begin{cases} 1 & \text{SE } N = 0 \\ N \times (N-1) \times (N-2) \times \ldots\ldots \times 3 \times 2 \times 1 & \text{SE } N > 0 \end{cases}$$

```
READ    0
STORECONST 1   1
STORECONST 1   2
JUMP   1    5
MUL      2    0
STORE   2
SUB        0    1
 STORE   0
JUMP      0  -4
WRITE   2
```

0	5
1	1
2	1

R	5
2	5
R	4
0	4

0	4
1	1
2	5

JUMP 1 5

MUL 2 0

STORE 2

SUB 0 1

STORE 0

JUMP 0 -4

WRITE 2

R	20

2	20

R	3

0	3

0	3
1	1
2	20

```
JUMP    1   5
MUL     2   0
STORE   2
SUB     0   1
STORE   0
JUMP    0  -4
WRITE   2
```

R	60

2	60

R	2

0	2

0	2
1	1
2	60

```
JUMP    1   5
MUL     2   0
STORE   2
SUB     0   1
STORE   0
JUMP    0  -4
WRITE   2
```

R	120

2	120

R	1

0	1

0	1
1	1
2	120

JUMP 1 5

MUL 2 0

STORE 2

SUB 0 1

STORE 0

JUMP 0 -4

WRITE 2

R	120

2	120

R	0

0	0

120

ARMAZENAMENTO DE VALORES NA MEMÓRIA DO NOSSO COMPUTADOR HIPOTÉTICO

VAMOS SUPOR QUE O TAMANHO DA NOSSA MEMÓRIA SEJA DE OITO ESPAÇOS, EM CADA ESPAÇO PODEMOS ARMAZENAR UM ALGARISMO DA BASE DECIMAL OU SEJA DE 0 A 9.

POR EXEMPLO, O NÚMERO ZERO SERIA REPRESENTADO POR:

0	0	0	0	0	0	0	0

CONSIDERANDO APENAS VALORES INTEIROS POSITIVOS PODERIAMOS ARMAZENAR 108 VALORES DISTINTOS E O MAIOR VALOR INTEIRO POSITIVO SERIA:**

9	9	9	9	9	9	9	9

COMO PODERÍAMOS ARMAZENAR TAMBÉM VALORES INTEIROS NEGATIVOS?

UMA FORMA SIMPLES DE SE FAZER ISSO É RESERVAR UM ESPAÇO DE MEMÓRIA DOS 8 EXISTENTES PARA ARMAZENAR O SINAL DO NÚMERO.

SE NESSE ESPAÇO RESERVADOR O VALOR ARMAZENADO FOR 0, O NÚMERO É POSITIVO SE FOR 1 O NÚMERO É NEGATIVO:

POR EXEMPLO O NÚMERO - 8345 SERIA REPRESENTADO NA NOSSA MÁQUINA POR:

1	0	0	0	8	3	4	5

E SE DESEJARMOS ARMAZENAR NÚMEROS REAIS?

EXISTEM DUAS MANEIRAS BASTANTE UTILIZADAS PARA ISSO:

PRIMEIRA:

> **RESERVAMOS UMA PARTE DAS 8 POSIÇÕES PARA ARMAZENAR A PARTE INTEIRA E UMA OUTRA PARTE PARA ARMAZENAR A PARTE FRACIONÁRIA DO NÚMERO EM QUESTÃO.**

COM AMBAS AS PARTES SEGUINDO A REPRESENTAÇÃO DE NÚMEROS INTEIROS JÁ VISTO ANTERIORMENTE.

VAMOS SUPOR QUE RESERVAMOS 5 POSIÇÕES PARA ARMAZENAR A PARTE INTEIRA DO NÚMERO, INCLUINDO O SEU SINAL, E AS TRÊS POSIÇÕES RESTANTES FICAM RESERVADAS PARA A PARTE FRACIONÁRIA.

O NÚMERO – 345.9 SERIA ENTÃO REPRESENTADO POR:

1	0	3	4	5	9	0	0

ESSE TIPO DE REPRESENTAÇÃO É CONHECIDA COM EM PONTO FIXO POIS O PONTO DECIMAL ESTÁ SEMPRE IMPLICITAMENTE REPRESENTADO NA MESMA POSIÇÃO.

LIMITAÇÕES: O MAIOR E O MENOR REAL QUE PODERIA SER ARMAZENADO NA MÁQUINA SERIA: +9999.999 E –9999.999

0	9	9	9	9	9	9	9
1	9	9	9	9	9	9	9

SEGUNDA MANEIRA:

ESSA MANEIRA DE REPRESENTAR NÚMEROS REAIS UTILIZA A NOTAÇÃO CIENTÍFICA A QUAL ESCREVERIA O NÚMERO – 345.9 ASSIM:

$$- 0.3459 \times 10^3$$

Reservamos então uma parte das posições para representar o expoente da base 10 (com sinal) e o restante para representar a mantissa (parte fracionária, com sinal).

VAMOS RESERVAR 5 POSIÇÕES PARA A MANTISSA E AS TRÊS RESTANTES PARA O EXPOENTE.

O NÚMERO:

– 345.9 QUE É IGUAL A - 0.3459 x 10^3

FICARIA ASSIM REPRESENTADO:

1	3	4	5	9	0	0	3

ESSE TIPO DE REPRESENTAÇÃO É CONHECIDA COM REPRESENTAÇÃO EM PONTO FLUTUANTE POIS A POSIÇÃO DO PONTO DECIMAL DO NÚMERO VARIA CONFORME O VALOR DO EXPOENTE.

LIMITAÇÕES:

PODEMOS AGORA REPRESENTAR VALORES REAIS QUE VARIAM

DE: + ou - 0.9999×10^{99}

ATÉ: + ou - 0.0001×10^{-99}

ERROS

1. SUPONHA QUE QUEREMOS MULTIPLICAR:

$0.23x10^{50}$ POR $-0.455x10^{60}$

0	2	3	0	0	0	5	0
1	4	5	5	0	0	6	0

QUAL SERIA O RESULTADO?

$$-0.10456 \times 10^{110}$$

ESSE VALOR NÃO PODERIA SER ARMAZENADO POIS O EXPOENTE É MAIOR DO QUE 99.

ESSE TIPO DE ERRO É CONHECIDO COMO

OVERFLOW

UM ERRO SIMILAR A ESSE É QUANDO O EXPOENTE ALCANÇA UM VALOR NEGATIVO QUE TAMBÉM NÃO POSSA SER REPRESENTADO, POR EXEMPLO, QUANTO TEMOS UM EXPONTE

EM 10^{-110}. NESSE CASO O VALOR É MUITO PRÓXIMO DE ZERO, MAS NÃO PODE SER PRECISAMENTE

ARMAZENADO NA MEMÓRIA.

ESSE TIPO DE ERRO É CONHECIDO COMO: **UNDERFLOW**

EXISTE AINDA UM OUTRO TIPO DE ERRO MAIS SUTIL E MUITO COMUM QUE OCORRE QUANDO O RESULTADO DE UMA UMA OPERAÇÃO É UM NÚMERO COM MAIS ALGARISMOS SIGNIFICATIVOS DO QUE SE PODE ARMAZENAR NA MEMÓRIA (ultrapassa o espaço reservado para a mantissa).

POR EXEMPLO O RESULTADO DA DIVISÃO DE 23.5 POR 8 QUE É 2.9375 SERIA REPRESENTADO NO NOSSO COMPUTADOR COMO:

0	2	9	3	7	0	0	1

QUE EQUIVALERIA A 2.937

ESSE ERRO É CONHECIDO COMO **TRUNCAMENTO** E QUANDO TRABALHAMOS COM CÁLCULOS NUMÉRICOS SOFISTICADOS DEVEMOS ESTAR CIENTES DA PERDA DE PRECISÃO NAS OPERAÇÕES REALIZADAS.