

Fundamentos de Sistemas Computacionais

Trabalho 2

Nomes: Leonardo do Nascimento Goulart, Gabriel Petry

Emails: leonardo.goulart003@edu.pucrs.br,
gabriel.petry@edu.pucrs.br

Primeira Solução:

main

```
ldi r1,string1
ldi r6,next_string
bnz r7,first_letter
```

next_string

```
ldi r1,string2
ldi r6,next_string1
bnz r7,first_letter
```

next_string1

```
ldi r1,string3
ldi r6,end
bnz r7,first_letter
```

end

hcf

first_letter

```
ldi r4,1
ldb r3,r1
ldi r2,0x20
sub r3,r3,r2
bez r3,decreases_one
bnz r7,loop
```

```

decreases_one
    sub r4,1
    bnz r7,loop

loop
    ldb r3,r1
    ldi r2,0x20
    sub r3,r3,r2
    bez r3,check_space
    ldb r3,r1
    add r1,1
    bnz r3,loop

print_all
    stw r4,0xf002
    ldi r4,0xa
    stw r4,0xf000
    bnz r7,r6

check_space
    add r1,1
    ldb r3,r1
    ldi r2,0x20
    sub r3,r3,r2
    bez r3,check_space

if_last
    ldb r3,r1
    bez r3,loop

increment_one
    add r4,1
    bnz r7,loop

end

    hcf

string1 "estou testando"
string2 "aa bcd ef  g h"
string3 "123 assembly  a"

```

main:

O Código começa carregando o endereço da primeira string, em seguida define o endereço do próximo código a ser executado após a contagem de palavras da primeira string. “first_letter” inicia a contagem de palavras.

next_string:

Carrega o endereço da segunda string, em seguida define o endereço do próximo código a ser executado no registrador r6, enfim se o registrador r7 for diferente de 0, “first_letter” será executado.

next_string1:

Carrega o endereço da terceira string, define o endereço do próximo código a ser executado (end) no registrador r6, enfim se o registrador r7 for diferente de 0, “first_letter” será executado.

first_letter:

Inicializa a contagem de palavras no registrador r4 com o valor 1, carrega a primeira letra da string no registrador r3 [carrega também o espaço no seu formato ASCII (0x20) no registrador r2], em seguinte subtrai o valor carregado em r2 do valor de r3, se r3 for igual a 0 (indicando que a primeira letra é um espaço), e então segue para “decrease_one”. Se r3 não for zero, o loop principal continuará processando a string.

decreases_one:

Decrementa a contagem de palavras carregadas no registrador r4 em 1, segue para o loop principal para continuar processando a string.

loop:

Carrega a letra atual da string no registrador r3, então subtrai o valor de r2 (0x20) do valor de r3, se r3 for zero, executa “check_space”, se r3 não for zero, incrementa o endereço no registrador r1, seguindo para a próxima letra, continuando o loop.

print_all:

Armazena o contador de palavras (registrador r4) no endereço 0xf002, então carrega o valor ASCII para nova linha (0x0a) no registrador r4 e armazena no endereço de saída, executa o próximo código definido em r6.

check_space:

Incrementa o endereço no registrador r1 para seguir para a próxima letra, carrega a próxima letra no registrador r3, então subtrai o valor de r2 do valor de r3, caso r3 seja zero, continua verificando espaços, se r3 for diferente de 0, executa "if_last".

if_last:

Carrega a letra atual no registrador r3, se o r3 for zero (final da string), volta para o loop principal, se r3 for diferente de 0, incrementa a contagem de palavras e retorna ao loop.

Increment_one:

Incrementa a contagem de palavras no registrador r4 em 1, volta para o loop principal.

end:

fim do programa.

Tabela de Simbolos:

```
0000 main
000e next_string
001c next_string1
002a first_letter
0040 decreases_one
0048 loop
0060 print_all
0072 check_space
0082 if_last
008a increment_one
0092 end
0094 string1
00a4 string2
00b4 string3
```

Código de Máquina:

0000	9900	ldc r1,0
0002	9994	ldc r1,148
0004	9e00	ldc r6,0
0006	9e0e	ldc r6,14
0008	9800	ldc r0,0
000a	982a	ldc r0,42
000c	d0e0	bnz r0,r7,r0
000e	9900	ldc r1,0
0010	99a4	ldc r1,164
0012	9e00	ldc r6,0
0014	9e1c	ldc r6,28
0016	9800	ldc r0,0
0018	982a	ldc r0,42
001a	d0e0	bnz r0,r7,r0
001c	9900	ldc r1,0
001e	99b4	ldc r1,180
0020	9e00	ldc r6,0
0022	9e92	ldc r6,146
0024	9800	ldc r0,0
0026	982a	ldc r0,42
0028	d0e0	bnz r0,r7,r0
002a	8c01	ldr r4,1
002c	0306	ldb r3,r0,r1
002e	9a00	ldc r2,0
0030	9a20	ldc r2,32
0032	6368	sub r3,r3,r2
0034	9800	ldc r0,0
0036	9840	ldc r0,64
0038	c060	bez r0,r3,r0
003a	9800	ldc r0,0
003c	9848	ldc r0,72
003e	d0e0	bnz r0,r7,r0
0040	6c01	sub r4,1
0042	9800	ldc r0,0
0044	9848	ldc r0,72
0046	d0e0	bnz r0,r7,r0
0048	0306	ldb r3,r0,r1
004a	9a00	ldc r2,0
004c	9a20	ldc r2,32
004e	6368	sub r3,r3,r2
0050	9800	ldc r0,0
0052	9872	ldc r0,114
0054	c060	bez r0,r3,r0
0056	0306	ldb r3,r0,r1
0058	5901	add r1,1
005a	9800	ldc r0,0
005c	9848	ldc r0,72

```

005e d060    bnz r0,r3,r0
0060 98f0    ldc r0,240
0062 9802    ldc r0,2
0064 5082    stw r0,r4,r0
0066 9c00    ldc r4,0
0068 9c0a    ldc r4,10
006a 98f0    ldc r0,240
006c 9800    ldc r0,0
006e 5082    stw r0,r4,r0
0070 d0f8    bnz r0,r7,r6
0072 5901    add r1,1
0074 0306    ldb r3,r0,r1
0076 9a00    ldc r2,0
0078 9a20    ldc r2,32
007a 6368    sub r3,r3,r2
007c 9800    ldc r0,0
007e 9872    ldc r0,114
0080 c060    bez r0,r3,r0
0082 0306    ldb r3,r0,r1
0084 9800    ldc r0,0
0086 9848    ldc r0,72
0088 c060    bez r0,r3,r0
008a 5c01    add r4,1

008c 9800    ldc r0,0
008e 9848    ldc r0,72
0090 d0e0    bnz r0,r7,r0
0092 0003    ???
0094 6573    ???
0096 746f    ???
0098 7520    ???
009a 7465    ???
009c 7374    ???
009e 616e    ???
00a0 646f    ???
00a2 0000    and r0,r0,r0
00a4 6161    sbc r1,r3,r0
00a6 2062    ???
00a8 6364    sub r3,r3,r1
00aa 2065    ???
00ac 6620    sub r6,r1,r0
00ae 2067    ???
00b0 2068    xor r0,r3,r2
00b2 0000    and r0,r0,r0
00b4 3132    ???
00b6 3320    slt r3,r1,r0
00b8 6173    ???

```

```

00ba 7365    ???
00bc 6d62    sub r5,98
00be 6c79    sub r4,121
00c0 2020    xor r0,r1,r0
00c2 6100    sub r1,r0,r0

```

Dump da memória:

```

0000: 9900 9994 9e00 9e0e 9800 982a d0e0 9900 |.....*....|
0010: 99a4 9e00 9e1c 9800 982a d0e0 9900 99b4 |.....*....|
0020: 9e00 9e92 9800 982a d0e0 8c01 0306 9a00 |.....*....|
0030: 9a20 6368 9800 9840 c060 9800 9848 d0e0 |. ch...@.`...H..|
0040: 6c01 9800 9848 d0e0 0306 9a00 9a20 6368 |l....H.....ch|
0050: 9800 9872 c060 0306 5901 9800 9848 d060 |...r.`...Y....H.`|
0060: 98f0 9802 5082 9c00 9c0a 98f0 9800 5082 |....P.....P.|
0070: d0f8 5901 0306 9a00 9a20 6368 9800 9872 |..Y.....ch...r|
0080: c060 0306 9800 9848 c060 5c01 9800 9848 |.`.....H.`\....H|
0090: d0e0 0003 6573 746f 7520 7465 7374 616e |....estou testan|
00a0: 646f 0000 6161 2062 6364 2065 6620 2067 |do..aa bcd ef  g|
00b0: 2068 0000 3132 3320 6173 7365 6d62 6c79 | h..123 assembly|
00c0: 2020 6100 0000 0000 0000 0000 0000 0000 |  a.....|

```

Segunda solução:

main

; carrega o vetor e o tamanho máximo e chama a função max_subarray

ldi r1,v

ldw r2,max

ldi lr,retmain1

bnz r7,max_subarray

retmain1

ldi r1,v1

ldw r2,max1

ldi lr,retmain2

bnz r7,max_subarray

retmain2

```
    ldi r1,v2
    ldw r2,max2
    ldi lr,end
    bnz r7,max_subarray
end
    hcf
```

```
max_subarray
    sub r7,2
    stw lr,r7

    sub r7,2
    stw 0,r7 ; espaco pro indice inicial
    sub r7,2
    stw 0,r7 ; espaco pro indice final

    ldi r3,0 ;i
    ldi r4,-1000 ; maior soma até o momento
    ldi r5,0 ; soma atual
    ldi r6,0
max_subarray_loop
    sub r7,2
    stw r1,r7

    slt r1,r3,r2
    bez r1,max_subarray_end

    ldw r1,r7
    add r7,2

    sub r7,2
    stw r2,r7

    ldw r2,r1
    add r1,2
```


add r5,r5,r2

slt r2,r4,r5

bnz r2,check_max_subarray

sub r7,2

stw r1,r7

ldi r1,0

slt r2,r5,r1

bnz r2,new_start

bnz r7,end_loop

check_max_subarray

ldw r2,r7

add r7,2

ldi r4,0

add r4,r4,r5

add r7,2

stw r6,r7

sub r7,2

stw r3,r7

sub r7,2

stw r2,r7

sub r7,2

stw r1,r7

ldi r1,0

slt r2,r5,r1

bnz r2,new_start

bnz r7,end_loop

new_start

ldi r5,0

ldi r6,1

add r6,r6,r3

end_loop

ldw r1,r7

add r7,2

ldw r2,r7

add r7,2

add r3,1

bnz r7,max_subarray_loop

max_subarray_end

ldw r1,r7

add r7,2

ldw r5,r7

add r7,2

ldw r1,r7

add r7,2

print_all

ldi r2,0x73

stw r2,0xf000

ldi r2,0x6f

stw r2,0xf000

ldi r2,0x6d

stw r2,0xf000

ldi r2,0x61

stw r2,0xf000

ldi r2,0x3a

stw r2,0xf000

stw r4,0xf002

ldi r2,0x0a

stw r2,0xf000

ldi r2,0x69

stw r2,0xf000

ldi r2,0x6e

stw r2,0xf000

ldi r2,0x69

stw r2,0xf000

ldi r2,0x63

stw r2,0xf000

ldi r2,0x69

stw r2,0xf000

ldi r2,0x6f

stw r2,0xf000

ldi r2,0x3a

stw r2,0xf000

stw r1,0xf002

ldi r2,0x0a

stw r2,0xf000

ldi r2,0x66

stw r2,0xf000

ldi r2,0x69

stw r2,0xf000

ldi r2,0x6e

stw r2,0xf000

ldi r2,0x61

stw r2,0xf000

ldi r2,0x6c

```
stw r2,0xf000
```

```
ldi r2,0x3a
```

```
stw r2,0xf000
```

```
stw r5,0xf002
```

```
ldi r2,0x0a
```

```
stw r2,0xf000
```

```
ldw r6,r7
```

```
add r7,2
```

```
bnz r7,r6
```

```
v    5 2 -2 -7 3 14 10 -3 9 -6 4 1
```

```
max   12
```

```
v1    5 -2 -3 7 -1 2 3 10 -4 2 -1 6
```

```
max1   12
```

```
v2   -320 555 120 30 -10 -5 10 -20 50
```

```
max2    9
```

Explicação:

Primeiramente, alguns registradores são inicializados: r3 como contador para percorrer os elementos da sequência, r4 armazena a maior soma encontrada até o momento, r5 a soma atual do subarray e r6 o índice de início do subarray atual.

No loop principal (max_subarray_loop), percorre-se cada elemento do vetor carregado. O valor atual do vetor é carregado e a soma atual (r5) é atualizada adicionando o valor atual do vetor. Após isso, verifica-se se a soma atual é maior que a maior soma já encontrada (r4). Caso seja, a maior soma é atualizada e os índices de início e fim são armazenados. Se a soma atual for negativa, a soma atual é reiniciada para zero e o índice de início do próximo subarray possível (r6) é atualizado. Dentro do loop, também há verificações para decidir se o loop deve continuar ou terminar. Por exemplo, se o índice atual (r3) é igual ao tamanho do vetor (r2), o loop termina.

Durante todo o processo, o ponteiro de pilha (r7) é utilizado para salvar o registrador lr e para armazenar temporariamente os índices de início e fim do subarray. Ao fim das operações, os valores são recuperados e o ponteiro de pilha é ajustado para liberar o espaço. Após completar o loop para um vetor, o programa imprime a maior soma (r4) e os índices de início (r1) e fim (r5) do subarray de soma máxima. Este processo se repete para três vetores diferentes, sendo que o programa retorna ao ponto de entrada (retmain, retmain1) após processar cada vetor.

Tabela de Símbolos:

main 0x0000
retmain1 0x0014
retmain2 0x0028
end 0x003c
max_subarray 0x003e
max_subarray_loop 0x0054
check_max_sub 0x008a
new_start 0x00b2
end_loop 0x00b8
max_subarray_end 0x00c8
print_all 00d4
v 0x01be
max 0x01d6
v1 0x01d8
max1 0x01f0
v2 0x01f2
max2 0x0204

Código de máquina:

0000	9901	ldc r1,1
0002	99be	ldc r1,190
0004	9801	ldc r0,1
0006	98d6	ldc r0,214
0008	4202	ldw r2,r0,r0
000a	9e00	ldc r6,0
000c	9e14	ldc r6,20
000e	9800	ldc r0,0
0010	983e	ldc r0,62
0012	d0e0	bnz r0,r7,r0
0014	9901	ldc r1,1
0016	99d8	ldc r1,216
0018	9801	ldc r0,1
001a	98f0	ldc r0,240
001c	4202	ldw r2,r0,r0
001e	9e00	ldc r6,0
0020	9e28	ldc r6,40
0022	9800	ldc r0,0
0024	983e	ldc r0,62
0026	d0e0	bnz r0,r7,r0
0028	9901	ldc r1,1
002a	99f2	ldc r1,242
002c	9802	ldc r0,2
002e	9804	ldc r0,4

0030	4202	ldw r2,r0,r0
0032	9e00	ldc r6,0
0034	9e3c	ldc r6,60
0036	9800	ldc r0,0
0038	983e	ldc r0,62
003a	d0e0	bnz r0,r7,r0
003c	0003	???
003e	6f02	sub r7,2
0040	50de	stw r0,r6,r7
0042	6f02	sub r7,2
0044	501e	stw r0,r0,r7
0046	6f02	sub r7,2
0048	501e	stw r0,r0,r7
004a	8b00	ldr r3,0
004c	8cfc	ldr r4,252
004e	9c18	ldc r4,24
0050	8d00	ldr r5,0
0052	8e00	ldr r6,0
0054	6f02	sub r7,2
0056	503e	stw r0,r1,r7
0058	3168	slt r1,r3,r2
005a	9800	ldc r0,0

005c	98c8	ldc r0,200
005e	c020	bez r0,r1,r0
0060	411e	ldw r1,r0,r7
0062	5f02	add r7,2
0064	6f02	sub r7,2
0066	505e	stw r0,r2,r7
0068	4206	ldw r2,r0,r1
006a	5902	add r1,2
006c	55a8	add r5,r5,r2
006e	3294	slt r2,r4,r5
0070	9800	ldc r0,0
0072	988a	ldc r0,138
0074	d040	bnz r0,r2,r0
0076	6f02	sub r7,2
0078	503e	stw r0,r1,r7
007a	8900	ldr r1,0
007c	32a4	slt r2,r5,r1
007e	9800	ldc r0,0
0080	98b2	ldc r0,178
0082	d040	bnz r0,r2,r0
0084	9800	ldc r0,0
0086	98b8	ldc r0,184
0088	d0e0	bnz r0,r7,r0
008a	421e	ldw r2,r0,r7
008c	5f02	add r7,2
008e	8c00	ldr r4,0
0090	5494	add r4,r4,r5
0092	5f02	add r7,2
0094	50de	stw r0,r6,r7
0096	6f02	sub r7,2
0098	507e	stw r0,r3,r7
009a	6f02	sub r7,2
009c	505e	stw r0,r2,r7
009e	6f02	sub r7,2
00a0	503e	stw r0,r1,r7
00a2	8900	ldr r1,0
00a4	32a4	slt r2,r5,r1
00a6	9800	ldc r0,0
00a8	98b2	ldc r0,178
00aa	d040	bnz r0,r2,r0
00ac	9800	ldc r0,0
00ae	98b8	ldc r0,184
00b0	d0e0	bnz r0,r7,r0
00b2	8d00	ldr r5,0
00b4	8e01	ldr r6,1
00b6	56cc	add r6,r6,r3

```

00b8 411e    ldw r1,r0,r7
00ba 5f02    add r7,2
00bc 421e    ldw r2,r0,r7
00be 5f02    add r7,2
00c0 5b01    add r3,1
00c2 9800    ldc r0,0
00c4 9854    ldc r0,84
00c6 d0e0    bnz r0,r7,r0
00c8 411e    ldw r1,r0,r7
00ca 5f02    add r7,2
00cc 451e    ldw r5,r0,r7
00ce 5f02    add r7,2
00d0 411e    ldw r1,r0,r7
00d2 5f02    add r7,2
00d4 9a00    ldc r2,0
00d6 9a73    ldc r2,115
00d8 98f0    ldc r0,240
00da 9800    ldc r0,0
00dc 5042    stw r0,r2,r0
00de 9a00    ldc r2,0
00e0 9a6f    ldc r2,111
00e2 98f0    ldc r0,240
00e4 9800    ldc r0,0

00e6 5042    stw r0,r2,r0
00e8 9a00    ldc r2,0
00ea 9a6d    ldc r2,109
00ec 98f0    ldc r0,240
00ee 9800    ldc r0,0
00f0 5042    stw r0,r2,r0
00f2 9a00    ldc r2,0
00f4 9a61    ldc r2,97
00f6 98f0    ldc r0,240
00f8 9800    ldc r0,0
00fa 5042    stw r0,r2,r0
00fc 9a00    ldc r2,0
00fe 9a3a    ldc r2,58
0100 98f0    ldc r0,240
0102 9800    ldc r0,0
0104 5042    stw r0,r2,r0
0106 98f0    ldc r0,240
0108 9802    ldc r0,2
010a 5082    stw r0,r4,r0
010c 9a00    ldc r2,0
010e 9a0a    ldc r2,10
0110 98f0    ldc r0,240

```


0112	9800	ldc r0,0
0114	5042	stw r0,r2,r0
0116	9a00	ldc r2,0
0118	9a69	ldc r2,105
011a	98f0	ldc r0,240
011c	9800	ldc r0,0
011e	5042	stw r0,r2,r0
0120	9a00	ldc r2,0
0122	9a6e	ldc r2,110
0124	98f0	ldc r0,240
0126	9800	ldc r0,0
0128	5042	stw r0,r2,r0
012a	9a00	ldc r2,0
012c	9a69	ldc r2,105
012e	98f0	ldc r0,240
0130	9800	ldc r0,0
0132	5042	stw r0,r2,r0
0134	9a00	ldc r2,0
0136	9a63	ldc r2,99
0138	98f0	ldc r0,240
013a	9800	ldc r0,0
013c	5042	stw r0,r2,r0
013e	9a00	ldc r2,0
0140	9a69	ldc r2,105
0142	98f0	ldc r0,240
0144	9800	ldc r0,0
0146	5042	stw r0,r2,r0
0148	9a00	ldc r2,0
014a	9a6f	ldc r2,111
014c	98f0	ldc r0,240
014e	9800	ldc r0,0
0150	5042	stw r0,r2,r0
0152	9a00	ldc r2,0
0154	9a3a	ldc r2,58
0156	98f0	ldc r0,240
0158	9800	ldc r0,0
015a	5042	stw r0,r2,r0
015c	98f0	ldc r0,240
015e	9802	ldc r0,2
0160	5022	stw r0,r1,r0
0162	9a00	ldc r2,0
0164	9a0a	ldc r2,10
0166	98f0	ldc r0,240
0168	9800	ldc r0,0
016a	5042	stw r0,r2,r0
016c	9a00	ldc r2,0

016e	9a66	ldc r2,102
0170	98f0	ldc r0,240
0172	9800	ldc r0,0
0174	5042	stw r0,r2,r0
0176	9a00	ldc r2,0
0178	9a69	ldc r2,105
017a	98f0	ldc r0,240
017c	9800	ldc r0,0
017e	5042	stw r0,r2,r0
0180	9a00	ldc r2,0
0182	9a6e	ldc r2,110
0184	98f0	ldc r0,240
0186	9800	ldc r0,0
0188	5042	stw r0,r2,r0
018a	9a00	ldc r2,0
018c	9a61	ldc r2,97
018e	98f0	ldc r0,240
0190	9800	ldc r0,0
0192	5042	stw r0,r2,r0
0194	9a00	ldc r2,0
0196	9a6c	ldc r2,108
0198	98f0	ldc r0,240
019a	9800	ldc r0,0
019c	5042	stw r0,r2,r0
019e	9a00	ldc r2,0
01a0	9a3a	ldc r2,58
01a2	98f0	ldc r0,240
01a4	9800	ldc r0,0
01a6	5042	stw r0,r2,r0
01a8	98f0	ldc r0,240
01aa	9802	ldc r0,2
01ac	50a2	stw r0,r5,r0
01ae	9a00	ldc r2,0
01b0	9a0a	ldc r2,10
01b2	98f0	ldc r0,240
01b4	9800	ldc r0,0
01b6	5042	stw r0,r2,r0
01b8	461e	ldw r6,r0,r7
01ba	5f02	add r7,2
01bc	d0f8	bnz r0,r7,r6
01be	0005	???
01c0	0002	ldb r0,r0,r0
01c2	ffffe	???
01c4	fff9	???
01c6	0003	???
01c8	000e	ldb r0,r0,r3

01ca	000a	ldb r0,r0,r2
01cc	fffd	???
01ce	0009	???
01d0	fffa	???
01d2	0004	and r0,r0,r1
01d4	0001	???
01d6	000c	and r0,r0,r3
01d8	0005	???
01da	fffe	???
01dc	fffd	???
01de	0007	???
01e0	ffff	???
01e2	0002	ldb r0,r0,r0
01e4	0003	???
01e6	000a	ldb r0,r0,r2
01e8	fffc	???
01ea	0002	ldb r0,r0,r0
01ec	ffff	???
01ee	0006	ldb r0,r0,r1
01f0	000c	and r0,r0,r3
01f2	fec0	???
01f4	022b	???

01f6	0078	and r0,r3,r6
01f8	001e	ldb r0,r0,r7
01fa	fff6	???
01fc	fffb	???
01fe	000a	ldb r0,r0,r2
0200	ffec	???
0202	0032	ldb r0,r1,r4
0204	0009	???

Dump da memória:

```

0000: 9901 99be 9801 98d6 4202 9e00 9e14 9800 |.....B.....|
0010: 983e d0e0 9901 99d8 9801 98f0 4202 9e00 |.>.....B...|
0020: 9e28 9800 983e d0e0 9901 99f2 9802 9804 |.(...>.....|
0030: 4202 9e00 9e3c 9800 983e d0e0 0003 6f02 |B....<...>...o.|
0040: 50de 6f02 501e 6f02 501e 8b00 8cfc 9c18 |P.o.P.o.P.....|
0050: 8d00 8e00 6f02 503e 3168 9800 98c8 c020 |....o.P>1h....|
0060: 411e 5f02 6f02 505e 4206 5902 55a8 3294 |A._.o.P^B.Y.U.2.|
0070: 9800 988a d040 6f02 503e 8900 32a4 9800 |.....@o.P>..2...|
0080: 98b2 d040 9800 98b8 d0e0 421e 5f02 8c00 |...@.....B._...|
0090: 5494 5f02 50de 6f02 507e 6f02 505e 6f02 |T._.P.o.P~o.P^o.|
00a0: 503e 8900 32a4 9800 98b2 d040 9800 98b8 |P>..2.....@....|
00b0: d0e0 8d00 8e01 56cc 411e 5f02 421e 5f02 |.....V.A._.B._.|
00c0: 5b01 9800 9854 d0e0 411e 5f02 451e 5f02 |[....T..A._.E._.|
00d0: 411e 5f02 9a00 9a73 98f0 9800 5042 9a00 |A_.....s....PB..|
00e0: 9a6f 98f0 9800 5042 9a00 9a6d 98f0 9800 |.o....PB...m....|
00f0: 5042 9a00 9a61 98f0 9800 5042 9a00 9a3a |PB....a....PB....|
0100: 98f0 9800 5042 98f0 9802 5082 9a00 9a0a |....PB....P.....|
0110: 98f0 9800 5042 9a00 9a69 98f0 9800 5042 |....PB...i....PB|
0120: 9a00 9a6e 98f0 9800 5042 9a00 9a69 98f0 |...n....PB...i..|
0130: 9800 5042 9a00 9a63 98f0 9800 5042 9a00 |..PB...c....PB..|
0140: 9a69 98f0 9800 5042 9a00 9a6f 98f0 9800 |.i....PB...o....|
0150: 5042 9a00 9a3a 98f0 9800 5042 98f0 9802 |PB....:....PB....|
0160: 5022 9a00 9a0a 98f0 9800 5042 9a00 9a66 |P".....PB...f|
0170: 98f0 9800 5042 9a00 9a69 98f0 9800 5042 |....PB...i....PB|
0180: 9a00 9a6e 98f0 9800 5042 9a00 9a61 98f0 |...n....PB...a..|
0190: 9800 5042 9a00 9a6c 98f0 9800 5042 9a00 |..PB...l....PB..|
01a0: 9a3a 98f0 9800 5042 98f0 9802 50a2 9a00 |.:....PB....P...|
01b0: 9a0a 98f0 9800 5042 461e 5f02 d0f8 0005 |.....PBF._.....|
01c0: 0002 fffe fff9 0003 000e 000a fffd 0009 |.....|
01d0: fffa 0004 0001 000c 0005 fffe fffd 0007 |.....|
01e0: ffff 0002 0003 000a fffc 0002 ffff 0006 |.....|
01f0: 000c fec0 022b 0078 001e fff6 fffb 000a |.....+.x.....|
0200: ffec 0032 0009 0000 0000 0000 0000 0000 |...2.....|

```