E-fólio B | Instruções para a realização do E-fólio

21010 – Arquitetura de Computadores

Enunciado

Pretende-se implementar no P3 um programa que identifique os primeiros números (de 16 bits), que pertencem ao conjunto S:

$$S = \{n \in [0,65535[: n \leq 2 \vee \exists_{m \in S} n \in \{(n\%m)^2 + 2 \cdot (n\%m), 2 \cdot (n\%m) + 1\}\}$$

Considere que o operador % é o resto da divisão inteira, e que este valor é nulo no caso do divisor ser 0.

Note que se *m* for maior que *n*, o resultado do resto da divisão é *n*, e portanto nunca passaria o teste para pertencer a *S*. Assim, é suficiente testar valores de *m* inferiores a *n*.

Deve implementar o solicitado em cada alínea, mediante alteração do programa em anexo. Em cada alínea deve implementar a sub-rotina solicitada, e no relatório mostrar o resultado da execução de acordo com o mostrado em anexo. Estes resultados são para os dados de entrada atuais, mas o programa deve funcionar para quaisquer outros dados de entrada.

a) **[1]** Pretende-se que implemente a sub-rotina ***TesteNM***, que recebe *n* em *R1* e *m* em *R2*, e identifica se $n \in \{(n\%m)^2 + 2 \cdot (n\%m), 2 \cdot (n\%m) + 1\}$. Retorna em *R1* o valor 1 no caso positivo, e 0 no caso contrário.

Casos de teste:

N	M	Passos - resultado
1	0	$n\%m = 1\%0 = 0 \| 1 \in \{(0)^2 + 2 \cdot (0), 2 \cdot (0) + 1\} \| 1 \in \{0,1\}$ - sim
2	1	$n\%m = 2\%1 = 0 \| 2 \in \{0,1\}$ - não
3	2	$n\%m = 3\%2 = 1 \| 3 \in \{(1)^2 + 2 \cdot (1), 2 \cdot (1) + 1\} \| 3 \in \{3,3\}$ - sim
9	7	$n\%m = 9\%7 = 2 \| 9 \in \{(2)^2 + 2 \cdot (2), 2 \cdot (2) + 1\} \| 9 \in \{8,5\}$ - não
15	4	$n\%m = 15\%4 = 3 \| 15 \in \{(3)^2 + 2 \cdot (3), 2 \cdot (3) + 1\} \| 15 \in \{15,7\}$ - sim

b) **[1]** Implemente a sub-rotina ***TesteN***, que recebe um valor *n* e indica se este pertence ao conjunto *S*, testando para tal todos potenciais números *m*, entre 0 e *n-1*, que pertençam a *S*. Notar que deve testar um dado número, apenas se este pertencer a S, pelo que deve chamar a função de teste recursivamente. Deve ser retornado no registo *R1* o valor de 1 no caso de pertencer a S, e 0 caso contrário. Considere que os números até 2 pertencem todos a S.

Nota: pretende-se nesta alínea a versão recursiva, que implementa a definição de modo direto. A versão iterativa e utilizando memória para guardar os resultados dos elementos menores (programação dinâmica), é solicitado na alínea C e D. Caso não consiga implementar esta sub-rotina recursivamente, passe para a alínea C, e retorne a esta alínea após realizar as restantes alíneas.

Casos de teste:

N	Passos - resultado
2	$n \leq 2$ – sim

4	TesteN(4):
	• TesteN(0)=1, TesteNM(4,0)=0
	• TesteN(1)=1, TesteNM(4,1)=0
	• TesteN(2)=1, TesteNM(4,2)=0
	• TesteN(3):
	  ○ TesteN(0)=1, TesteNM(3,0)=0
	  ○ TesteN(1)=1, TesteNM(3,1)=0
	  ○ TesteN(2)=1, TesteNM(3,2)=1, TesteNM(4,3)=0 - não

5	TesteN(5):
	• TesteN(0)=1, TesteNM(5,0)=0
	• TesteN(1)=1, TesteNM(5,1)=0
	• TesteN(2)=1, TesteNM(5,2)=0
	• TesteN(3):
	  ○ TesteN(0)=1, TesteNM(3,0)=0
	  ○ TesteN(1)=1, TesteNM(3,1)=0
	  ○ TesteN(2)=1, TesteNM(3,2)=1, TesteNM(5,3)=1 – sim

6	TesteN(6):
	• TesteN(0)=1, **TesteNM(6,0)**=0
	• TesteN(1)=1, **TesteNM(6,1)**=0
	• TesteN(2)=1, **TesteNM(6,2)**=0
	• TesteN(3):
	  ○ TesteN(0)=1, TesteNM(3,0)=0
	  ○ TesteN(1)=1, TesteNM(3,1)=0
	  ○ TesteN(2)=1, TesteNM(3,2)=1, **TesteNM(6,3)**=0
	• TesteN(4):
	  ○ TesteN(0)=1, TesteNM(4,0)=0
	  ○ TesteN(1)=1, TesteNM(4,1)=0
	  ○ TesteN(2)=1, TesteNM(4,2)=0
	  ○ TesteN(3):
	  ▪ TesteN(0)=1, TesteNM(3,0)=0
	  ▪ TesteN(1)=1, TesteNM(3,1)=0
	  ▪ TesteN(2)=1, TesteNM(3,2)=1, TesteNM(4,3)=0
	• TesteN(5):
	  ○ TesteN(0)=1, TesteNM(5,0)=0
	  ○ TesteN(1)=1, TesteNM(5,1)=0
	  ○ TesteN(2)=1, TesteNM(5,2)=0
	  ○ TesteN(3):
	  ▪ TesteN(0)=1, TesteNM(3,0)=0
	  ▪ TesteN(1)=1, TesteNM(3,1)=0
	  ▪ TesteN(2)=1, TesteNM(3,2)=1, TesteNM(5,3)=1, **TesteNM(6,5)**=0 - não

O caso de teste com N=4, todos os valores inferiores pertencem a S, pelo que acabam por ser testados. No caso de teste com N=6, o valor de 4 não é testado dado que não pertence a S. Para clarificar, são colocados a verde os testes feitos para o número 6, todos com apenas valores de *m* que pertencem a S.

c) **[1]** O anterior procedimento é muito lento. Pretende-se que faça a sub-rotina ***TesteVetorN***, de forma iterativa que preencha um vector com os números pertencentes a *S*, de 0 a *N-1*. Deve fazer uso dos resultados já calculados, começando pelos elementos menores, já que para calcular um elemento, é suficiente ter o resultado de todos os elementos menores. Em *R1* é colocado está o valor de *N*, e em *R2* está o início do vetor onde devem ser colocados os resultados: colocar 0 se não pertence a *S* e 1 se pertence.

Nota: não deve reutilizar a alínea B, já que é recursiva. Deve utilizar os valores colocados no próprio vetor para saber se um dado elemento *m* pertence a *S*.

Nota2: pode utilizar como casos de teste, os valores da alínea B, pelo que os primeiros valores no vetor (números de N de 0 a 6) são: 1, 1, 1, 1, 0, 1, 0.

Exemplo do processamento para N=6:
TesteVetorN(6):

- vetor[0]=1, TesteNM(6,0)=0
- vetor[1]=1, TesteNM(6,1)=0
- vetor[2]=1, TesteNM(6,2)=0
- vetor[3]=1, TesteNM(6,3)=0
- vetor[4]=0
- vetor[5]=1, TesteNM(6,5)=0
- vetor[6] <- 0

Os valores anteriores já estavam calculados quando se começou a calcular N=6, e no final colocou-se no vetor o valor de *TesteVetorN(6)*, de modo a ser utilizado em futuros cálculos sem ser necessário recalcular.

d) **[1]** O anterior procedimento utiliza muita memória. Pretende-se que faça a sub-rotina ***TesteVetorB*** de forma idêntica à alínea C, mas utilizando apenas um bit por cada número. Assim, cada posição de memória guarda 16 bits, e também 16 números. Em *R1* é colocado está o valor de *N*, e em *R2* está o início do vetor onde devem ser colocados os resultados de forma binária.

Nota: não deve nesta alínea alocar um vetor expandido, e converter no final para binário. Deve utilizar para memória apenas o vetor binário fornecido, no caso dos dados de entrada do exemplo, 64 posições de memória, para guardar 1024 valores binários. A versão que utiliza o número de posições de memória igual à quantidade de números processados, é avaliada na alínea C.

Nota2: pode utilizar como casos de teste os valores das alíneas anteriores, e sabendo que o número 7 não pertence a *S*, os 8 primeiros bits, que são: `2fh`

Programa de teste:

```
                ; zona de colocação de dados entrada/saida
                ORIG 8000h
testeN          STR     1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
testeM          STR     0,1,2,1,2,3,4,0,7,5, 2, 3, 4, 1, 4, 5
resultadosA     TAB     16
resultadosB     TAB     16
resultadosC     TAB     64
resultadosD     TAB     64
testes          EQU     16
testesC         EQU     64
testesD         EQU     1024

                ; zona do código
                ORIG 0000h

                ; inicialização do stack
Inicio:         MOV R1, fd1fh
                MOV SP, R1

                ; TESTE alinea A
                MOV R3, testes
CicloA:         DEC R3
                            BR.N FimA
                MOV R1, M[R3+testeN]     ; R1 - enviado valor de N
                MOV R2, M[R3+testeM]     ; R2 - enviado valor de M
                PUSH R3                  ; manter a variável iteradora
                CALL TesteNM             ; chamada ao procedimento solicitado
                POP R3
                MOV M[R3+resultadosA],R1 ; guardar resultado retornado
                BR CicloA
FimA:           Nop

                ; TESTE alinea B
                MOV R2, testes
CicloTesteN:    DEC R2
                BR.N FimB
                MOV R1, M[R2+testeN]     ; R1 - enviado valor de N
                PUSH R2                  ; manter a variável iteradora
                CALL TesteN              ; chamada ao procedimento solicitado
                POP R2
                MOV M[R2+resultadosB],R1 ; guardar resultado retornado
                BR CicloTesteN
FimB:           Nop


                ; TESTE alinea C
                MOV R1, testesC
                MOV R2, resultadosC
                CALL TesteVetorN
FimC:           Nop

                ; TESTE alinea D
                MOV R1, testesD
                MOV R2, resultadosD
                CALL TesteVetorB
FimD:           JMP Fim

                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ; função TesteNM solicitada na alinea A
                ; Entrada: R1 - valor de N; R2 - valor de M
                ; Saída: R1 - resultado
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
TesteNM:        MOV R1, R0
                RET

                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ; função TesteN solicitada na alinea B
                ; Entrada: R1 - valor de N
                ; Saída: R1 - resultado
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
TesteN:         MOV R1, R0
                RET


                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ; função TesteVetorN solicitada na alinea C
                ; Entrada: R1 - valor máximo de N; R2 - vetor
                ; Saída: vetor em R2 corretamente preenchido
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
TesteVetorN:    RET


                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ; função TesteVetorB solicitada na alinea D
                ; Entrada: R1 - valor máximo de N; R2 - vetor binário
                ; Saída: vetor em R2 corretamente preenchido
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
TesteVetorB:    RET

                ; manter a última instrução intacta
Fim:            JMP Fim
```

Resultado de execução do programa de teste, sem o código de qualquer alínea:

```
Clock: 3256
Instructions: 357
8000 : 0001 0002 0003 0004 0005 0006 0007 0008 ........
8008 : 0009 000a 000b 000c 000d 000e 000f 0010 ........
8010 : 0000 0001 0002 0001 0002 0003 0004 0000 ........
8018 : 0007 0005 0002 0003 0004 0001 0004 0005 ........
8020 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8028 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8030 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8038 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8040 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8048 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8050 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8058 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8060 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8068 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8070 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8078 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8080 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8088 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8090 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
8098 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
80a0 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
80a8 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
80b0 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
80b8 : 0000 0000 0000 0000 0000 0000 0000 0000 ........
```

BOM TRABALHO!

Avaliação

Cotação:
A cotação encontra-se junto de cada uma das alíneas, entre [].

Critérios de Correcção:
Funcionalidade: 50%
Simplicidade e Modularidade: 10%
Eficiência (serão contabilizados o número de instruções e ciclos de relógio): 10%
Apresentação do código (indentação e comentários): 20%
Relatório (Legibilidade e Justificação dos Resultados e das Opções): 10%

Descontos:
Trabalhos entregues que não estejam em conformidade com as regras de entrega do e-fólio B: até 10%
Código sem comentários, ou apenas com comentários a reflectir o significado da instrução (exemplo MOV R1,R2 ;mover o conteúdo de R2 para R1) : até 50%
Detecção de fraude (total ou parcial): 100%
Trabalhos entregues após a data limite (máximo 24h) : 10%

Regras para entrega do e-fólio B:

Forma de entrega:

Deverá ser entregue um relatório em formato pdf ou Word até 5 páginas A4, com todos os cálculos e todas as opções tomadas na construção dos programas. Em anexo deve colocar todo o código (apenas o código das rotinas) e resultados obtidos (número de instruções, ciclos de relógio e conteúdo da memória de 8000h a 80bfh). O código deve estar num formato que permita a seleção de modo a ser copiado e colado para o simulador do P3.

Não são aceites entregas fora da plataforma Moodle.