6.	Em 1945, von Neumann definiu um conjunto de conceitos sobre a construção de uma máquina de cálculo universal, que se passou a designar por "Máquina de von Neumann". Esses conceitos podem ser resumidos em 3 contribuições fundamentais: (1) uma organização para esta máquina; (2) uma arquitectura (conjunto de instruções); (3) o conceito de "stored program". Caracterize brevemente cada uma destas contribuições.
	instruções), (o) o conocito de storea program : caracterize preveniente cada uma destas contribuições.
_	
-	
-	
7.	Indique os vários estágios de execução da instrução beq \$t2, \$t3, -12, explicitando os elementos do datapath single cycle do MIPS utilizados (ver figura anexa), as operações realizadas em cada fase, e indique o valor dos sinais de controlo RegDst, RegWrite, ALUSrc, PCSrc, MemWrite, MemRead e MemToReg.

## **PRÁTICA**

A questão 8 vale 4 valores, a questão 9 vale 2,5 valores e a questão 10 vale 2 valores.

**8.** Considere o bloco de programa escrito em linguagem C que calcula a nota média numa lista de alunos.

```
P8a:___
P8b:___
P8c:___
P8d:___
Σ:___
```

```
typedef struct{
    char nome[22];
    int nota;
}aluno;

aluno alunos[128];

int NotaMedia(aluno * lstalunos){
    int i,soma;
    soma=0;
}for (i=0;i<128;i++){
        soma+=lstalunos[i].nota;
    }
    return (int)(soma/128);
}</pre>
```

(a) Complete o programa em baixo, nas linhas marcadas de 1 a 6, de modo a corresponder a uma compilação válida do programa em C.

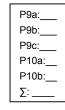
```
NotaMedia:
                                              0(,%eax,4), %edx
                                        leal
                                              8(%ebp), %eax
     pushl %ebp
                                        movl
     movl %esp, %ebp
                                              24(%edx,%eax), %edx
                                        movl
                                             -8(%ebp), %eax
           $12, %esp
      subl
                                        leal
     movl $0, -8(%ebp)
                                  .L8:
/*1*/ movl $0, _____
                                  /*3*/ addl
                                              %edx, _
.L6:
                                                      _, %eax
                                  /*4*/ leal
      cmpl $127, -4(%ebp)
                                        incl
                                              (%eax)
/*2*/ jg
                                  /*5*/ jmp
.L7:
                                  .L9:
                                  /*5*/ movl
                                                      __, %eax
      movl
           -4(%ebp), %edx
                                        sarl $7, %eax
      movl
            %edx, %eax
      sall
           $3, %eax
                                        leave
      subl %edx, %eax
                                        ret
```

(b) Identifique e explique as instruções responsáveis pelo teste do ciclo for.

(c) Identifique e explique as instruções responsáveis pelo cálculo da soma: soma+=lstalunos[i].nota
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.
(d) Que alterações seria necessário fazer, no programa assembly, caso a lista de alunos passasse a ter 256 alunos.

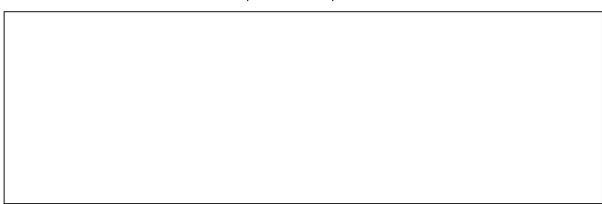
**9.** Considere o seguinte fragmento de código, escrito em assembly do MIPS:

```
$a1, $t0, $t1
    slt
             $s0, $s1, $L1
    ble
             $fp, $sp
    move
             $t2, 12($s1)
    sw
             $a0, -4($fp)
    lw
$L1:
    sll
             $s0, $s1, 2
                              # QB
    li
             $v0, 0xFA00FF
```



a) Reescreva o código fornecido, usando apenas instruções nativas do MIPS.

b) Converta a instrução assinalada com **"# QB"** para código máquina do MIPS, apresentando o resultado final em hexadecimal e incluindo na resposta todos os passos intermédios usados.



c) Converta para assembly do MIPS a instrução máquina **0x A171FFFA** (em hexadecimal), apresentando os cálculos intermédios.

10. Considere uma máquina com uma frequência do relógio de 1.5 GHz e uma cache organizada em blocos de 1 palavra. O tempo de acesso à memória central é composto por uma latência de 20 ns mais 10 ns por cada palavra. A máquina necessita, em média, de 2.58 ciclos para executar cada instrução, sem incluir os acessos à memória central. Esta máquina é usada para executar um programa com 109 instruções (20% das quais implicam um acesso a dados em memória) e exibe uma *miss rate* de instruções de 6%. a) Se o tempo de execução deste programa for de 4s, qual a miss rate de dados? b) Para tirar partido da localidade espacial, o tamanho dos blocos da cache foi aumentado para 4 palavras. Esta alteração resultou numa diminuição da miss rate de instruções para 4% e de dados para 6%. Qual o ganho relativamente às condições da alínea anterior? Como justifica este resultado?