PROBLEM 3:

a) 100%/(10%+90%/32) = 100/12,81 = 7,8
Bei 32 Prozessoren kann das Programm 7,8 mal so schnell ausgeführt werden.

b) 100%/(10%+90%/n) =n_to_infty=> 100/10 = 10
Das Programm kann bei unendlich vielen Prozessoren maximal 10 mal so schnell ausgeführt werden.

PROBLEM 1:

a) Bei 1-Bit Sprungvorhersage wird immer vorausgesagt, dass sich der nächste Sprung so wie der letzte verhält.

b) Bei 2-Bit Sprungvorhersage wird zwischen schwacher und starker Vorhersage unterschieden. Wenn die schwache vorhersage einmal richtig war, geht man in den Zustand für starke Vorhersage. Im Falle von if Blöcken in einer Schleife, ist dies günstiger (siehe c3).

c)

(1)

```
MOV RCX, 1
for:
    CMP RCX, n
    JG end

        ...

    INC RCX ;i++
    JMP for
end:
```

1-Bit:
beim ersten mal falsche Vorhersage
beim letzten mal falsche Vorhersage

also n-2 richtige und 2 falsche Vorhersagen
-lim-> 100% richtig

2-Bit:
beim ersten und zweiten mal falsche Vorhersage
beim letzten mal falsche Vorhersage

also n-3 richtige und 3 falsche Vorhersagen
-lim-> 100% richtig

(2)

```
MOV RCX, 1 ;i=1
for1:
    CMP RCX, n
    JG end1 ;end if i>n

    MOV RDX, 1 ;j=1
    for2:
        CMP RDX, n
        JG end2 ;end if j>n

            ...

        INC RDX
        JMP for2 ;j++
    end2:

    INC RCX ;i++
    JMP for1
end1:
```

1-Bit:
jede i-Vorhersage bis auf die letzte falsch (1 von n richtig)
jede letzte j-Vorhersage falsch (n*(n-1) von n^2 richtig)

(n^2-n+1)/(n^2+n)
-lim-> 100% richtig

2-Bit
erste und letzte i-Vorhersage falsch, sonst richtig (n-2 von n richtig)
aller erste j-Vorhersage und jede letzte j-Vorhersage falsch und sonst richtig (n*(n-1)-1 von n^2 richtig)

(n^2-3)/(n^2+n)
-lim-> 100% richtig

(3)

```
MOV RCX, 1
for:
    CMP RCX, n
    JG end

    CMP p, 1
    JNE not_p ;if p do block

        ...

    not_p:

    INC RCX ;i++
    JMP for
end:
```

1-Bit:
erste i-Vorhersage falsch und letzte mit 50% falsch (also 1.5 falsche)
solange p-wahr ist, gibt es keine falschen Vorhersagen.
sobald p-falsch ist, gibt es eine falsche p und eine falsche i Vorhersage.

(0.5*n+(0.5*n-1.5))/(2*n) = (n-1.5)/(2*n)
-lim-> 50%

2-Bit:
erste i-Vorhersage falsch.
angenommen p ist beim ersten mal wahr (ansonsten wird die Rechnung zu kompliziert). Dann ist die erste p-Vorhersage falsch. Danach ist die i-Vorhersage immer richtig und die p-Vorhersage mit 50% richtig. Die letzte i-Vorhersage ist falsch.

(n-2+0.5*n)/(2*n) = (1.5*n-2)/(2*n)
-lim-> 75% richtig

PROBLEM 2:

a)

```
INC a       |IF|ID|OF|EX|WB|
INC a       |  |  |  |IF|ID|OF|EX|WB|
INC a       |  |  |  |  |  |  |IF|ID|OF|EX|WB|
CMP x,0     |  |  |  |  |  |  |  |IF|ID|OF|EX|WB|
JNE no_if   |  |  |  |  |  |  |  |  |IF|ID|OF|EX|WB|
SHR a,1     |  |  |  |  |  |  |  |  |  |IF|ID|OF|EX|WB|
INC a       |  |  |  |  |  |  |  |  |  |  |  |  |IF|ID|OF|EX|WB|
no_if:      |  |  |  |  |  |  |  |  |  |  |  |  |  |PF|PF|PF|PF|PF|PF|PF|PF|PF|PF|
INC a       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |IF|ID|OF|EX|WB|
INC a       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |IF|ID|OF|EX|WB|
            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |    31 Takte
```

b)

```
INC a       |IF|ID|OF|EX|WB|
INC a       |  |  |  |IF|ID|OF|EX|WB|
INC a       |  |  |  |  |  |  |IF|ID|OF|EX|WB|
CMP x,0     |  |  |  |  |  |  |  |IF|ID|OF|EX|WB|
CSHRE a,1   |  |  |  |  |  |  |  |  |IF|ID|OF|EX|WB|
CINCE a     |  |  |  |  |  |  |  |  |  |  |IF|ID|OF|EX|WB|
INC a       |  |  |  |  |  |  |  |  |  |  |  |  |IF|ID|OF|EX|WB|
INC a       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |IF|ID|OF|EX|WB|
            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |    20 Takte
```