

# Memorial Descritivo

## 1º Trabalho AOC1

---

### **P1) REVERSE ENGINEERING**

A metodologia adotada para a resolução desse problema foi primeiramente converter todo o código hexadecimal em questão para IAS Assembly, para poder compreender o código. Após, o código foi interpretado passo a passo descrevendo a ideia do algoritmo em um arquivo. Ao fim, depois de várias verificações na interpretação, foi descrito um código em alto nível (Linguagem C) e assim finalizada a compreensão do código em questão.

Resumo do código:

- O código divide sucessivamente o valor da posição 100 até que ele se torne zero.
- Se antes de dividir, o número da posição 100 for par, o valor da posição 101 é acumulado na posição 102.
- Após essa verificação o valor da posição 100 é dividido por 2 e é atualizado, e o valor da posição 101 é multiplicado por 2 e é atualizado.
- Assim sucessivamente até que o valor da posição 100 seja 0.

**Obs:** A descrição objetiva se encontra no arquivo P1.

### **P2) OPTIMIZATION**

O problema apresentava uma operação a ser aplicada (potenciação) bem como dois métodos para sua realização. O primeiro passo foi transcrever os códigos em Linguagem C para a linguagem da IAS. Feito isso, os programas foram levados a teste.

Para o teste, os programas foram executados automaticamente, rodando a 1 instrução por segundo, e foram cronometrados. Os valores utilizados foram os mesmos em ambos os testes:  $x = 3$  e  $n = 9$  (3 elevado à nona).

---

No teste do primeiro programa (sem otimização), o tempo gasto foi de 3 minutos e 7 segundos. Já no segundo programa (com otimização), o tempo registrado foi de 2 minutos e 19 segundos.

### **P3) PROGRAM CHALLENGE**

O código em c separa em vetores pares e ímpares um vetor de números inteiros. Para começar foi “declarado” as variáveis e constantes em endereços distantes do começo do programa, em seguida a codificação em ias das instruções e o uso um pouco mais complicado do `store (x,8:19)/store (x,28:39)` que é como um “ponteiro”.

### **P4) SORTING**

Para este problema, o ponto de início adotado foi usar o código dado na questão como referência para convertê-lo em IAS Assembly. Após vários testes e reestruturação do código, o código IAS Assembly foi codificado e um selection sort foi implementado. Ao fim, o mesmo foi convertido em hexadecimal.

### **P5) PROGRAM CHALLENGE**

Para esse problema utilizamos 2 laços sendo um deles interno e outro externo, em que a cada passo se decrementa 1 da variável de controle que é inicializada com (tamanho do vetor -1), assim na hora de fazer o salto condicional basta passar o valor da variável de controle no `ac` e testar com `jump+m`, para fazer a potência foi utilizado o código da questão 2 dando um salto incondicional, como se fosse uma função do código. Para testes foi utilizado o simulador IAS Machine da Unicamp.