

Nome: _____ RA: _____

1ª Prova

1. Analise o código C abaixo e responda as questões a seguir.

```
1. #include <stdio.h>
2. #include <fcntl.h>
3. int main(){
4.     int a, s, i, c;
5.     char b[64];
6.     char n1[20], n2[20];
7.     c = 65;
8.     gets(n1);
9.     gets(n2);
10.    a = open(n1, O_RDONLY | O_CREAT, 0755);
11.    s = open(n2, O_RDWR | O_APPEND | O_CREAT, 0777);
12.    while ((i = read(a, b, sizeof(b))) > 0){
13.        write(s, b, i);
14.        write(STDOUT, b, i);
15.    }
16.    lseek(s, 0L, SEEK_SET);
17.    write(s, &c, sizeof(c));
18.    close(a);
19.    close(s);
20. }
```

- a) (0,5) Explique o que acontece (em termos das variáveis n1, n2, a e s) quando as linhas 10 e 11 são executadas.
- b) (0,5) Explique o que acontece (em termos das variáveis i, a, b e s) quando o laço codificado nas linhas 12 a 15 é executado.
- c) (0,5) Explique o que acontece (em termos das variáveis s e c) quando as linhas 16 e 17 são executadas.
2. (1,0) (a) Quantos acessos seriam feitos em média para se encontrar uma determinada chave usando-se busca sequencial em um arquivo em disco contendo 250.000 registros? (b) Se esse mesmo arquivo fosse acessado em blocos de 50 registros cada, quantos acessos seriam necessários para se descobrir que uma dada chave não se encontra no arquivo?
3. Considere o arquivo de dados abaixo, com a seguinte organização:
- Registros de tamanho variável, precedidos por um campo de 2-bytes que armazena o tamanho do registro;
 - Campos delimitados pelo separador “|” (com exceção do campo de tamanho);
 - Cabeçalho com 16-bytes.

LED.Head: -1

0...	16...	98...	135...	186...
Cab...	80Sousa ...	35Melo ...	49Batista ...	61Silva ...

Considere o uso de uma *LED*, com retorno de espaço à lista após inserções. Simule as operações seguintes, mostrando, em cada passo, como fica o arquivo e o ponteiro *LED.Head*. Utilize a estratégia *worst-fit* (pior ajuste) na manipulação da *LED*. Inicialmente, *LED.head* = -1.

- a. (0,25) Remoção do 1o registro (chave “Sousa”)
- b. (0,25) Remoção do 2o registro (chave “Melo”)
- c. (0,25) Remoção do 3o registro (chave “Batista”)
- d. (0,25) Inserção de um novo registro de 40 bytes (chave “Alves”)
- e. (0,25) Inserção de um novo registro de 39 bytes (chave “Valdez”)
- f. (0,25) Inserção de um novo registro de 45 bytes (chave “Bastos”)

4. (1,5) Dado o *arquivo de dados* abaixo, monte dois índices secundários, um por Nacionalidade e outro por Profissão, usando o mesmo arquivo de *lista invertida* para os dois índices. Considere a Matrícula como chave primária e monte também o índice primário.

RRN	Matrícula	Profissão	Nacionalidade
0	2050	Analista	Chilena
1	430	Programador	Brasileira
2	980	Digitador	Argentina
3	1010	Operador	Brasileira
4	2000	Digitador	Paraguaia
5	1900	Analista	Brasileira
6	1550	Digitador	Chilena
7	690	Operador	Chilena
8	730	Programador	Brasileira
9	1100	Operador	Argentina
10	1790	Analista	Argentina
11	1990	Analista	Chilena
12	2200	Operador	Brasileira
13	1620	Programador	Brasileira
14	790	Digitador	Chilena
15	1040	Digitador	Argentina

5. (1,5) Considerando um arquivo de 5 GB, um *buffer* de entrada de 4 MB e um *buffer* de saída de 1 MB, calcule o custo (em n° de *seeks* e bytes transmitidos) de cada fase do *Merge Sort* em múltiplos passos, quando realizado em dois passos: primeiro 25 x 50-vias + 25-vias.