

Universidade Federal do ABC
MCTA028-15 - Programação Estruturada
2018.Q3

Lista de Exercícios 6

Professores Emílio Francesquini e Carla Negri Lintzmayer

27 de novembro de 2018

1. O que será impresso pelo programa abaixo:

```
#include <stdio.h>

struct T {
    int x;
    int y;
};

typedef struct T T;

void f1(T *a);

void f2(int *b);

int main() {
    T a, b, *c, *d;
    c = &a;
    a.x = 2;
    a.y = 4;
    b.x = 2;
    b.y = 2;
    d = c;
    f1(d);
    b = *d;

    printf("x: %d --- y: %d\n", b.x, b.y);
}
```

```

void f1(T *a) {
    f2(&(a->x));
    f2(&(a->y));
}

void f2(int *b) {
    *b = 2*(*b);
}

```

2. [PF] Defina um registro empregado para armazenar os dados (nome, sobrenome, data de nascimento, CPF, data de admissão, salário) de um colaborador de sua empresa. Defina um vetor de empregados para armazenar todos os colaboradores de sua empresa.
3. Faça um programa que armazene informações dos alunos matriculados na disciplina de PE (no máximo 100 alunos). Para cada aluno, queremos manter seu nome, RA, nota da prova 1, nota da prova 2, média das listas de exercícios, nota do projeto e quantidade de faltas. Seu programa deve imprimir um menu com as seguintes opções:
 - (a) Atualizar faltas de um aluno
 - (b) Atualizar nota da prova 1 de um aluno
 - (c) Atualizar nota da prova 2 de um aluno
 - (d) Atualizar média das listas de exercícios de um aluno
 - (e) Atualizar nota do projeto de um aluno
 - (f) Calcular a média final antes da recuperação de um dado aluno
 - (g) Verificar se um aluno pode realizar a prova de recuperação
4. Considere o seguinte registro:

```

struct dma {
    int dia, mes, ano;
};

```

Escreva uma função que receba duas variáveis do tipo `struct dma`, cada uma já representando uma data válida, e devolva o número de dias que decorreram entre as duas datas.

5. [PF] Escreva uma função que receba um número inteiro que representa um intervalo de tempo medido em minutos e devolva o número equivalente de horas e minutos (por exemplo, 131 minutos equivalem a 2 horas e 11 minutos).

Use uma struct como a seguinte:

```

struct hm {
    int horas;
    int minutos;
};

```

6. [PF] Um racional é qualquer número da forma p/q , sendo p inteiro e q inteiro não nulo. É conveniente representar um racional por um registro:

```

typedef struct {
    int p, q;
} racional;

```

Vamos convencionar que o campo q de todo racional é estritamente positivo e que o máximo divisor comum dos campos p e q é 1. Escreva funções:

- **reduz**, que receba inteiros a e b e devolva o racional que representa a/b ;
- **neg**, que receba um racional x e devolva o racional $-x$;
- **soma**, que receba racionais x e y e devolva o racional que representa a soma de x e y ;
- **produ**, que receba racionais x e y e devolva o racional que representa o produto de x por y ;
- **quoci**, que receba racionais x e y e devolva o racional que representa o quociente de x por y .

7. [PF] Escreva um programa que conte o número de ocorrências de cada um dos caracteres de um arquivo. O seu programa deve receber o nome do arquivo via linha de comando e imprimir uma tabela com o número de ocorrências de cada caractere do arquivo. Para ganhar inspiração, analise o comportamento do comando **wc** (*word count*), presente em todo sistema UNIX e GNU/Linux.
8. Dado um arquivo de nome **cadastros** que contém registros com os campos **nome**, **sexo**, **altura**, **peso** e **data-de-nascimento**, separá-lo em dois arquivos: um chamado **cadastros-homens** com registros cujo campo **sexo** apresente o valor **m**, e outro chamado **cadastros-mulheres** com registros cujo campo **sexo** apresente o valor **f**. Os registros dos novos arquivos não devem conter o campo **sexo**.
9. Escreva um programa que le tudo o que for escrito na entrada padrão (**stdin**) e que grava a saída em um arquivo cujo nome tenha sido passado na linha de comando. Veja o exemplo de execução abaixo:

```
$ meuProg arquivo.saida
```

Tudo o que **for** escrito aqui será escrito no arquivo cujo nome foi passado como parâmetro para o programa.

Ele deve aceitar ler múltiplas linhas e múltiplos

espaços.

Para entrar com um EOF no terminal basta digitar Ctrl+D

```
$
```

10. Escreva um programa que copia um arquivo para outro, equivalente ao comando `cp` do Linux. Atenção: seu programa deve ser capaz de trabalhar com arquivos de qualquer tamanho e com arquivos binários.