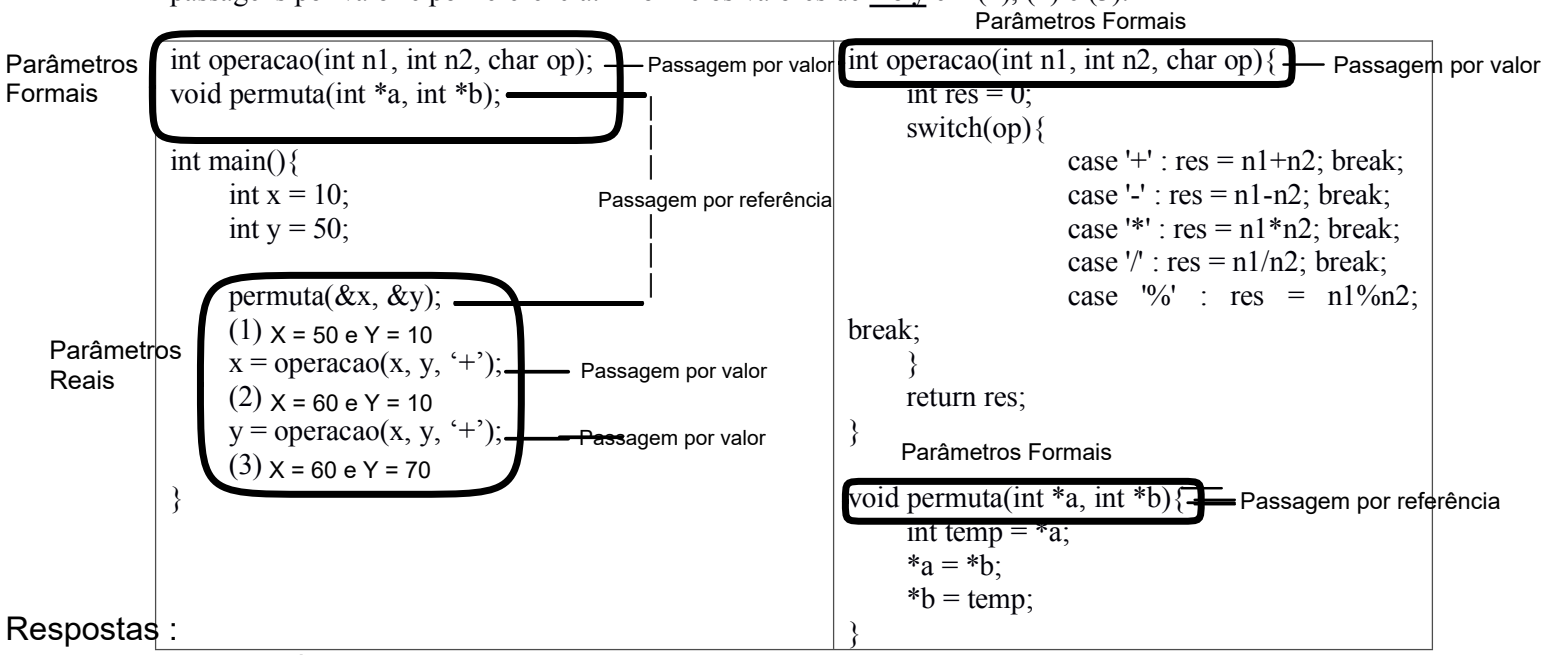
IFPB – Campus Campina Grande	
Disciplina: Laboratório de Estrutura de Dados	Turma: Engenharia de Computação
Professor: Francisco Dantas Nobre Neto	
Aluno: Jardel Brandon de Araujo Regis	Matrícula: 201621250014

PROVA 1

Leia com atenção as observações abaixo:

* Você poderá resolver esta prova em dupla, e informar o nome e a matrícula da dupla, no envio da prova;
* Qualquer material poderá ser consultado, porém, sugere-se que seja consultar um material confiável. Por exemplo, o material de C (da UFMG) e os slides do Professor Francisco Dantas (ambos disponíveis na mesma pasta compartilhada desta prova);
* Ao concluir a prova, você deverá enviá-la (anexo) por e-mail ao Professor Francisco Dantas (dantas.nobre@academico.ifpb.edu.br), com o assunto do e-mail *Prova 01 - C*, e no corpo o nome completo e a matrícula da dupla;
* Esta prova vale 10 pontos.
* **O prazo para entrega desta prova é a data 12/07/2017, às 23:59.**

1) O que são e qual a diferença entre os parâmetros reais e os formais? E a diferença entre passagem por valor e por referência? Identifique no trecho de código abaixo os parâmetros reais e formais, e as passagens por valor e por referência. Informe os valores de **x e y** em (1), (2) e (3).




```
int operacao(int n1, int n2, char op);
void permuta(int *a, int *b);

int main(){
    int x = 10;
    int y = 50;

    permuta(&x, &y);
    (1) X = 50 e Y = 10
    x = operacao(x, y, '+');
    (2) X = 60 e Y = 10
    y = operacao(x, y, '+');
    (3) X = 60 e Y = 70
}
```

```
int operacao(int n1, int n2, char op){
    int res = 0;
    switch(op){
            case '+' : res = n1+n2; break;
            case '-' : res = n1-n2; break;
            case '*' : res = n1*n2; break;
            case '/' : res = n1/n2; break;
            case '%' : res = n1%n2;
break;
    }
    return res;
}

void permuta(int *a, int *b){
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

Respostas :

Passagem por referência – É passada para a função uma referência da variável, sendo possível alterar o conteúdo da variável original usando-se esta referência. Na linguagem C a passagem por referência é implementada com o uso de ponteiros que apontam à posição de memória onde a variável está armazenada.

Passagem por valor – permite usar dentro de uma função uma cópia do valor de uma variável, porém não permite alterar o valor da variável original (somente a cópia pode ser alterada).

Os parâmetros da função na sua declaração são chamados parâmetros formais.

Na chamada da função os parâmetros são chamados parâmetros atuais/reais.

Ou seja os parâmetros formais (variáveis locais a função chamada) são inicializados com o valor dos parâmetros reais.

2) Uma string é uma palavra se não contiver caracteres em branco. Dado um texto como entrada no teclado:

a. Determinar a maior palavra e imprimi-la.
b. Classificar as palavras em 3 classes e determinar a frequência absoluta de ocorrência delas em cada classe. Cada classe será criada de acordo com o tamanho das palavras, da seguinte forma: classe 1 vai de 0 a 3 caracteres na palavra; classe 2 vai de 4 a 6 caracteres; e classe 3 a partir de 7 caracteres na palavra. Após, imprimir a ocorrência da seguinte forma:

Classe das palavras	Frequência
0 --- 3	10
4 --- 6	8
A partir de 7 (7 ou mais)	12

OBS.: O tamanho da variável que recebe os caracteres do teclado deve ser de 200, mas, para teste utilizar um valor menor.

3) Faça uma função em C que receba dois vetores como parâmetros, com tamanho máximo 5, e que retorne o quantitativo de números iguais entre os dois vetores. Exemplo: A = {1, 2, 3, 4, 5} e B = {1, 4, 5, 8, 10}. A função deverá retornar o valor 3, pois três números são comuns a ambos os vetores.

4) Faça uma função que receba dois parâmetros: uma string; e um vetor de inteiros. A função deverá armazenar, na primeira posição do vetor de inteiros, o quantitativo de letras maiúsculas. Na segunda posição, o quantitativo de letras minúsculas e, na terceira, o quantitativo de espaços em branco.

5) Criar uma estrutura de usuário, com os campos nome (char 30), login (char 10) e senha (char 10). Deverão ser criados, previamente, dois usuários (estaticamente) e, ao executar o programa, deverá ser solicitado o login e senha de quem rodou o programa. Se o login e senha for de algum dos dois usuários previamente cadastrado, o programa deverá apresentar uma mensagem “Seja bem-vindo Fulano”, em que Fulano é o nome do usuário. Caso o usuário não exista, informar a mensagem “Usuário não cadastrado!”, e sair do programa.

6) Crie uma união que possua dois campos: idade(int) e CPF(char[12]). Crie uma estrutura chamada Empregado, com os campos matricula(char[10]), nome(char[50]) e número(a união criada). Se a matrícula do usuário iniciar com “123”, significa que o campo da união a ser impresso é a idade, caso contrário, será impresso o CPF. Faça um programa que receba três empregados, de modo que **ocorra as duas possibilidades**, tanto matrículas com início “123”, como com valores diferentes. Se o início da matrícula do usuário for “123”, deverá ser solicitada idade, senão, o CPF. Você deverá imprimir todas as informações ao término do programa corretas.

7) Seja uma estrutura para descrever os imóveis de uma imobiliária, contendo os seguintes campos:
bairro: string de tamanho 20
preço: real
tamanho privativo: real
vagas de garagem: inteiro
tipo do imóvel: enumeração (apartamento ou casa)

a) Escrever a definição da estrutura imóvel.
b) Declarar o vetor vetor_imovel do tipo da estrutura definida acima, de tamanho 5 e local à função main( ).
c) Definir um bloco de programa para ler o vetor vetor_imovel.
d) Definir um bloco de programa que obtenha a string “apartamento” ou “casa”, quando os tipos dos imóveis forem, respectivamente, apartamento ou casa.
e) Defina um bloco de programa imprima o tipo do imóvel, seguido pelo bairro e preço.
OBS.: Um bloco de programa representa um conjunto de comandos, escritos em C, que está no corpo da função main( ).