

Para cada exercício listado abaixo, implemente as funções que devem ser criadas e o código principal contendo a função `main()` para a correta execução do programa.

Sugestão de software Codeblocks v13.12

Lista de Exercícios

1) Implemente uma função que receba dois valores reais, efetue a soma desses dois valores e retorne o resultado para o usuário.

2) Implemente uma função que verifique se um número é primo ou não, 1 se o número é primo, 0 se o número não é primo, conforme o protótipo:

```
int primo (int n);
```

3) Implemente uma função que imprima todos os números da série de Fibonacci até o valor informado à variável `n`. A série é dada por: 1, 1, 2, 3, 5, 8, 13, 21..., isto é, os dois primeiros termos são iguais a 1 e cada termo seguinte é a soma dos dois termos anteriores. Implemente conforme o protótipo:

```
void imprime_fibonacci (int n);
```

4) Implemente uma função que receba uma string como parâmetro e retorne como resultado o número de vogais nessa string, obedecendo o protótipo:

```
int numero_de_vogais (char * str);
```

5) Implemente uma função que receba uma string como parâmetro e altere os caracteres minúsculos para maiúsculos, de acordo com o protótipo:

```
void minusculo (char *str);
```

6) Implemente uma função que receba como parâmetros um vetor de números inteiros (`vet`) de tamanho `n` e retorne quantos números pares estão armazenados nesse vetor, seguindo o protótipo:

```
int pares (int n, int *vet);
```

7) Implemente uma função que receba como parâmetro um vetor de números inteiros (vet) de tamanho n (criado dinamicamente) e que inverta a ordem dos elementos armazenados nesse vetor (elemento da primeira posição vai para a última, o segundo elemento vai para a penúltima posição, e assim por diante...), conforme o protótipo:

```
void inverte (int n, int *vet);
```

8) Implemente uma função que aloque dinamicamente uma matriz de inteiros, passando como parâmetros o número de linhas e o número de colunas e imprima na tela os elementos armazenados e sua respectiva posição [lin x col], de acordo com os protótipos:

```
float **cria_matriz (int lin, int col);
```

```
void imprime (int lin, int col, float **mat);
```

9) Implemente um cadastro de alunos de uma disciplina e codifique as funções de cadastro do aluno, alteração do aluno e impressão dos alunos, utilizando a seguinte estrutura:

```
struct aluno {  
    char nome [81];  
    char matricula [8];  
    char turma;  
    float nota1;  
    float nota2;  
    float nota3;  
};  
typedef struct aluno Aluno;
```

Obs: Procurem realizar todos os exercícios, pois as avaliações serão baseadas a partir deles.