



Lista de exercícios – Ponteiros 02 - Vetores

Implemente a função descrita nas questões a seguir, seguindo as especificações. No mesmo exercício, construa a função main() que utilize ou teste a função desenvolvida.

1. void inverte(char *str);
Função : Inverte a ordem dos caracteres na string str
2. int vetor pares (int *vetor, int tamanho, int *p);
Função : Copia os valores pares de vetor para p.
Parametros : p – ponteiro para vetor de numeros pares
vetor – ponteiro para vetor com todos os numeros
tamanho – numero de itens de vetor
Retorno : quantidade de valores copiados para p.
DICA : declare variável para armazenar os dados apontados por p no programa principal.
3. void par_impar(int *vetor, int tamanho, int *par, int *npar, int *impar, int *nimpar);
Função : Copia os valores pares de vetor para par e valores impares para impar
Parametros : vetor – ponteiro para variavel de entrada de dados
tamanho – numero de itens de vetor
par – ponteiro para vetor que armazena valores pares
*npar – numero de valores copiados para par
impar – ponteiro para vetor que armazena valores impares
*nimpar – numero de valores copiados para par

DICA : declare variável para armazenar os dados apontados por par e impar no programa principal.
4. void ordena_c(int *dados, int tam);
Função : Ordena em ordem crescente os valores apontados por dados
Parametros : dados – ponteiro para vetor a ser ordenado
tamanho – numero de itens de dados
DICA : Utilize um algoritmo de ordenação conhecido, como o bubble sort ou o quick-sort
5. void ordena(int *dados, int tam, char ordem);
Função : Ordena em ordem crescente os valores apontados por dados
Parametros : dados – ponteiro para vetor a ser ordenado
tamanho – numero de itens de dados
ordem – 'c' para ordenamento crescente, 'd' para ordenamento decrescente.
Case insensitive – não importa se é maiuscula ou minuscula
Padrão é ordenamento crescente.
DICA : Utilize um algoritmo de ordenação conhecido, como o bubble sort ou o quick-sort