Exercícios com registros/tipos definido pelo usuário

01) Considere a estrutura abaixo, faça um programa que leia dois pontos cartesianos (x e y) e apresente a distância entre os pontos. Crie a função distancia2d que recebe dois parâmetros (estrutura) e devolva um double.

```
struct _PONTO {
  double x,y;
};
```

Ponteiros

02) Faça um programa que apresente todas as permutações de uma String dada usando ponteiros.
Exemplo: abcd

```
abcd  abdc  acbd  acdb  adcb  adbc  bacd  badc  bcad  bcda
bdca  bdac  cbad  cbda  cabd  cadb  cdab  cdba  db
ca  dbac  dcba  dcab  dacb  dabc
```

03) Faça um programa que leia N e um vetor de N floats e na sequencia encontre o maior número de um vetor alocado dinamicamente.

04) Faça um programa que leia uma String e passe a String para uma função que deverá contar o número de vogais e consoantes e apresente as quantidades. Use passagem por referência para os contadores de vogais e consoantes.

05) Faça uma função que apresente uma String de forma espelhada (do último símbolo para o primeiro). Use ponteiros para percorrer a String.

06) Faça uma função que apresente uma String de forma espelhada (do último símbolo para o primeiro). Use ponteiros para percorrer a String.

07) Faça uma função checkInterno que recebe os parâmetros ptr1, ptr2 e tam. A função deve retornar verdadeiro se ptr1 estiver na região entre ptr2 e ptr2+tam.

08) Faça um programa que leia uma lista de nomes, a leitura será terminada com "entrada vazia" (strlen == 0, str[0]=='\0'). Para cada nome lido aloque uma CEL em memória e coloque em uma lista dinâmica. No final, apresente a lista lida e libere corretamente os elementos de memória.

09) Faça um programa que leia uma frase (com até 512 bytes e não precisa validar) e apresente todos os sufixos da frase (use ponteiro e incremento de ponteiros, não use índices de vetor)
Ex: ao ser lido "casa" o programa deverá gerar:
casa
asa
as

s

10) Crie a função findStr que recebe dois parâmetros s1 e s2 e retorne o ponteiro da sequência de s1 onde inicia e contém s2.

11) Crie a função findStrIgnoreCase que recebe dois parâmetros s1 e s2 e retorne o ponteiro da sequência de s1 onde inicia e contém s2, ignorando variação de caixa (maiúsculas e minúsculas) entre s1 e s2.