

Algoritmos e Estruturas de Dados I

2018.Q1, Matutino

Lista de Exercícios 2: Ponteiros e estruturas dinâmicas

Profa. Mirtha Lina Fernández Venero

25 de fevereiro de 2018

1 Vetores, Ponteiros e Strings

1. Quais são os valores de x, y, z, p e pp ao final da função abaixo?

```
3 void f(){
4     int x, y, z, *p, **pp;
5     y = 0;
6     p = &y;
7     pp = &p;
8     x = *p;
9     x += 4;
10    (*p)++;
11    --x;
12    (*p) += x;
13    p--;
14    (*p) += x;
15    p++;
16    (**pp) += 3*x;
17 }
```

2. Identifique e corrija os erros nas funções abaixo.

```
19 // Item a)
20 ▾ void trocaMin(int *i, int *j) {
21     if (i < j) return;
22     int *temp;
23     *temp = *i;
24     *i = *j;
25     *j = *temp;
26 }
27
28 // Item b)
29 ▾ void main() {
30     char *a = "trem", *b = "carro";
31     if (a < b)
32         printf ("%s vem antes de %s no dicionário\n", a, b);
33     else
34         printf ("%s vem depois de %s no dicionário\n", a, b);
35 }
36
37 // Item c)
38 ▾ void xxx(char *v, int n, int k) {
39     char *c;
40     for (c = v; c < v + n; v += k)
41         printf ("%c", *c);
42 }
```

3. Reflita sobre os erros do seguinte programa (<https://repl.it/@mirthalina/arrayvsstring>) e como poderiam ser corrigidos.

```
1  #include <stdio.h>
2
3  int main() {
4      char *s1 = "Linguagem C";
5      char s2[ ] = "Linguagem C";
6      printf("%s\n", s1); printf("%s\n", s2);
7      printf(s1);
8      printf(s2);
9      printf(s1+2);
10     printf(*(s1+2));
11     printf(&s2[2]);
12     printf(s2+2);
13     printf(s2[2]);
14     printf(s1[2]);
15     s1 +=4;
16     s1++;
17     s2 +=4;
18     s2++;
19
20     char *s3 = "Linguagem", s4[ ], s5[20];
21     s3 = s4;
22     s2 = s5;
23     s5 = "Linguagem C";
24     s5 = s3;
25
26     return 0;
27 }
```

2 Operações sobre vetores dinâmicos e strings

Escreva funções C para realizar as seguintes operações. Pense cuidadosamente quais os parâmetros de entrada-saída e o retorno de cada função. Escreva um programa para testar suas funções.

1. Inverter um vetor (sem criar um novo).
2. Inserir um elemento na posição i do arranjo. Caso o valor de i exceda o número de elementos do vetor (porém não o tamanho) insira na posição i , preencha com zero os

elementos intermediários e atualize o número de elementos. Caso o valor de i exceda o tamanho do vetor, crie (se possível) um novo vetor com tamanho apropriado e insira como explicado anteriormente.

3. Dividir um vetor em dois vetores contendo os elementos negativos e positivos (não incluir os zeros).
4. Concatenar duas strings.
5. Dada uma string contendo uma frase (palavras separadas por espaço), obter outra com a conversão a maiúsculas, minúsculas ou título dependendo dum parâmetro de entrada.
6. Dada uma string e uma letra, retornar um vetor de inteiros contendo as posições das ocorrências da letra na string.