



Algoritmos e Estruturas de Dados I

2019.Q1, Matutino

Lista de Exercícios 1: Revisão da Linguagem C

Profa. Mirtha Lina Fernández Venero

Prof. Paulo Henrique Pisani

15 de fevereiro de 2019

Escreva programas C para resolver os seguintes problemas definindo funções apropriadas.

1 Intervalo inteiro de tamanho k

Dado um inteiro $k > 1$, gerar uma sequência de M números inteiros e determinar se ela contém pelo menos uma sub-sequência de até k números consecutivos. Faça o programa imprimir todas as sub-sequências máximas

Exemplo 1: Para $k = 4$, a sequência 8,-10,4,-2,-1,0,1,2,50,51,54 contém as sub-sequências

- -2,-1,0,1
- -1,0,1,2
- 50,51

Exemplo 2: Para $k = 7$, a sequência 5, 20, -9, 51, 52, 53, -1, -4, -3, -2, -1, 0, 1, 2, 3 contém as sub-sequências

- 51, 52, 53
- -4, -3, -2, -1, 0, 1, 2
- -3, -2, -1, 0, 1, 2, 3

2 Agenda de Contatos

Escreva um programa para gerenciar uma agenda de contatos. Para cada contato deve-se armazenar **Nome**, **Telefone** e **Email**. Seu programa deve dar as seguintes opções ao usuário: **Inserir um contato** no final da agenda, **Buscar dados pelo Nome**, **Buscar dados pelo Telefone**, **Mostrar Agenda** e **Sair**.

3 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 }
```

4. O seguinte gráfico mostra uma parte do FLUXOGRAMA DAS RECOMENDAÇÕES ENTRE DISCIPLINAS DO BCC. Escreva um programa que construa uma estrutura no heap para armazenar o gráfico abaixo onde todos nós tenham o mesmo tipo.

