UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

Departamento de Informática e Matemática Aplicada

Introdução às Técnicas de Programação — IMD0012 — 14 de setembro de 2017

◁ Exercícios - parte 7 ▷

1. Em relação ao setor de memória na Figura da página a seguir (um fictício com endereços de 16 bits):

   (a) Qual o endereço base de 3.2 e qual o do número inteiro 503?

   (b) O que é escrito na tela com printf("%s\n", p0);?

   (c) O que é escrito na tela com printf("%s\n", &p0[5]);?

   (d) O que é escrito na tela com printf("%s\n", p0+2);?

   (e) Qual o valor de *p1?

   (f) Qual o valor de p1[0]?

   (g) Qual o valor de p1+1?

   (h) Qual o valor de *(p1+1)?

   (i) Qual o valor de *(p2+1)?

   (j) Qual o valor de &x?

   (k) Qual o valor de &p0?

0x85ba	0	1	1	0	0	0	1	1	**char** c
0x85bb	0	1	1	0	0	0	0	1	**char** a
0x85bc	0	1	1	1	0	0	1	1	**char** s
0x85bd	0	1	1	0	0	0	0	1	**char** a
0x85be	0	0	0	0	0	0	0	0	**char** \0
0x85bf	0	1	1	0	0	0	0	1	**char** a
0x85c0	0	1	1	0	1	1	0	0	**char** l
0x85c1	0	1	1	0	1	0	0	1	**char** i
0x85c2	0	0	0	0	0	0	0	0	**char** \0
0x85c3	0	1	0	0	0	0	0	0	**float** 3.2
0x85c4	0	1	0	0	1	1	0	0	
0x85c5	1	1	0	0	1	1	0	0	
0x85c6	1	1	0	0	1	1	0	1	
0x85c7	0	1	0	0	0	0	0	0	**float** 5.4
0x85c8	1	0	1	0	1	1	0	0	
0x85c9	1	1	0	0	1	1	0	0	
0x85ca	1	1	0	0	1	1	0	1	
0x85cb	0	0	1	1	1	1	1	1	**float** 1.2
0x85cc	1	0	0	1	1	0	0	1	
0x85cd	1	0	0	1	1	0	0	1	
0x85ce	1	0	0	1	1	0	1	0	
0x85cf	0	1	0	0	0	0	0	1	**float** 8.4
0x85d0	0	0	0	0	0	1	1	0	
0x85d1	0	1	1	0	0	1	1	0	
0x85d2	0	1	1	0	0	1	1	0	
0x85d3	1	0	0	0	0	1	0	1	**char *p0 =** 0x85ba
0x85d4	1	0	1	1	1	0	1	0	
0x85d5	0	0	0	0	0	0	0	0	**int x =** 503
0x85d6	0	0	0	0	0	0	0	0	
0x85d7	0	0	0	0	0	0	0	1	
0x85d8	1	1	1	1	0	1	1	1	
0x85d9	0	1	1	0	0	1	0	0	**char** d
0x85da	0	0	0	0	0	0	0	0	**int** 702
0x85db	0	0	0	0	0	0	0	0	
0x85dc	0	0	0	0	0	0	1	0	
0x85dd	1	0	1	1	1	1	1	0	
0x85de	1	0	0	0	0	1	0	1	**float *p1 =** 0x85c3
0x85df	1	1	0	0	0	0	1	1	
0x85e0	1	0	0	0	0	1	0	1	**float *p2 =** 0x85cb
0x85e1	1	1	0	0	1	0	1	1	
0x85e2	1	0	0	0	0	1	0	1	**int *p3 =** 0x85ca
0x85e3	1	1	0	0	1	0	1	0	

Figura 1: Setor da memória para a primeira questão

2. ▷ Altere o ponteiro p para apontar para o início da segunda palavra da string. Assuma que a frase digitada possui ao menos uma palavra.

```
1
2 #include <stdio.h>
3
4 int main() {
5
6     char frase[200];
7
8     gets(frase);
9
10    char *p = frase;
11
12    //altere p para que aponte para o primeiro caractere apos o primeiro espaco
13
14    printf("%s\n", p);
15
16    return 0;
17 }
```

3. ▷ Altere o valor inicial de p de forma que a saída do programa seja 5 11 12 2 8.

```
1
2 #include <stdio.h>
3
4 int main() {
5
6     int i;
7     int v[] = {3, 14, 9, 6, 5, 11, 12, 2, 8, 13, 7, 10, 1, 4};
8
9     int *p; //atribua um valor inicial adequado
10
11    for(i = 0; i < 5; i++) {
12        printf("%d ", p[i]);
13    }
14
15    return 0;
16 }
```

4. ▷ Escreva um programa que leia um inteiro **n**, leia **n números reais** e escreva na tela o índice (começando de 1) do maior entre esses **n** números reais. Assuma que não há números iguais na sequência.

   Exemplo:

   6
   3.97 2.15 13.97 12.38 10.65 16.19
   4

5. ▷ O MEC precisa de sua ajuda (de novo!) para automatizar a correção das provas objetivas do ENEM. Mas dessa vez o MEC não tem ideia do número máximo de questões que pode haver. Escreva um programa que leia um número inteiro **n** representando o número de questões. Em seguida leia as **n** respostas do gabarito e, em seguida, as **n** respostas do aluno. Assuma que as respostas estão sempre entre 1 e 5. Depois o programa deve escrever na tela quantas questões o aluno acertou e a string "acertos" ou "acerto" (para 1 acerto), conforme exemplo abaixo.

   Exemplo 2:
   ```
   4
   1 2 3 4
   1 5 3 5
   2 acertos
   ```

   Exemplo 2:
   ```
   7
   1 2 3 2 1 5 4
   3 3 3 3 3 3 3
   1 acerto
   ```

6. ▷ Escreva um programa que leia um inteiro **n**, leia **n** inteiros que estão em ordem crescente, um inteiro **m** e **m** inteiros que também estão em ordem crescente. O programa deve em seguida escrever na tela uma única sequência ordenada com os **m+n** inteiros.

   Exemplo:
   ```
   4
   1 2 3 4
   5
   4 6 7 9
   1 2 3 4 4 6 7 9
   ```