

# 3o Pré-teste de Fundamentos de Programação, 24/11/2015

3a PROVA DE FUNDAMENTOS DE PROGRAMAÇÃO - Curso: Engenharia de Computação

3a

Data: 18/09/2013

Horário limite: 15:30

Resolva as questões abaixo identificando-as claramente na folha de respostas. Mantenha o silêncio na sala (mantendo desligado aparelhos eletrônicos). A interpretação das questões faz parte da prova.

(5 pontos) 1) Clareza e organização da folha de respostas.

(5 pontos) 2) Apresente a definição do conceito de Algoritmo.

~~(30 pontos) 3) Sobre manipulação de bits na linguagem C. Implemente uma função que realize o giro de n bits a esquerda sobre um dado número inteiro (1000000000). Protótipo da função: int rol (int x, int n); Exemplo: rol(1000000000, 4) = 4.  $0_310_30\dots b_5b_4b_3b_2b_1b_0 \rightarrow b_27b_26\dots b_1b_0b_31b_30b_29b_28$~~

(30 pontos) 4) Considerando o programa abaixo, o que será **impresso na saída padrão** ao executá-lo?

```
1 #include <stdio.h>
2 #define x 1
3 short w;
4 struct valor {
5     int *end;
6     char car;
7 };
8
9 int sub2(int *v, int *p, int n) {
10     int *a, *b, *c;
11     a = b = c = &n;
12     *a += x;
13     v[*p] = n;
14     printf ("%d %d\n", v[(a)+(b)+(c)], v[*p]);
15     return (*p);
16 }
17
18 void sub1(int *v, int *p, int n) {
19     int a, b;
20     a = n;
21     v[a] = *p;
22     *p = 0;
23     w = sub2(v, p, a);
24     b = w;
25     w = n + x;
26     w = sub2(v, p, b);
27 }
28
29 int main () {
30     int vet[10] = {2,4,4,4,2,2,1,1,0,0};
31     struct valor v;
32     w = 257;
33     v.end = &vet[0];
34     v.car = (char)w;
35     w = (short)v.car;
36     printf ("%d %d %d\n", *v.end, v.car, w);
37     sub1 (vet, v.end, v.car);
38     printf ("%d %d %d %d\n", vet[0], *v.end, v.car, w);
39     return 0;
40 }
```

(30 pontos) 5) Robô Colecionador (Maratona 2010, 1ª Fase Brasil): Um dos esportes favoritos na Robolândia é o Rali dos Robôs. Este rali é praticado em uma arena retangular gigante de N linhas por M colunas de células quadradas. Algumas das células estão vazias, algumas contêm figurinhas da Copa (muito

apreciadas pelas inteligências artificiais da Robolândia) e algumas são ocupadas por pilastras que sustentam o teto da arena. Em seu percurso os robôs podem ocupar qualquer célula da arena, exceto as que contém pilastras, que bloqueiam o seu movimento. O percurso do robô na arena durante o rali é determinado por uma sequência de instruções. Cada instrução é representada por um dos seguintes caracteres: ‘D’, ‘E’ e ‘F’, significando, respectivamente, “gire 90 graus para a direita”, “gire 90 graus para a esquerda” e “ande uma célula para a frente”. O robô começa o rali em uma posição inicial na arena e segue fielmente a sequência de instruções dada (afinal, eles são robôs!). Sempre que o robô ocupa uma célula que contém uma figurinha da Copa ele a coleta. As figurinhas da Copa não são repostas, ou seja, cada figurinha pode ser coletada uma única vez. Quando um robô tenta andar para uma célula onde existe uma pilastra ele patina, permanecendo na célula onde estava, com a mesma orientação. O mesmo também acontece quando um robô tenta sair da arena. Dados o mapa da arena, descrevendo a posição de pilastras e figurinhas, e a sequência de instruções de um robô, você deve escrever um programa para determinar o número de figurinhas coletadas pelo robô.

**Entrada:** A entrada contém vários casos de teste. A primeira linha de um caso de teste contém três números inteiros  $N$ ,  $M$  e  $S$  ( $1 \leq N, M \leq 100, 1 \leq S \leq 5 \times 10^4$ ), separados por espaços em branco, indicando respectivamente o número de linhas e o número de colunas da arena e o número de instruções para o robô. Cada uma das  $N$  linhas seguintes da entrada descreve uma linha de células da arena e contém uma cadeia com  $M$  caracteres. A primeira linha que aparece na descrição da arena é a que está mais ao Norte; a primeira coluna que aparece na descrição de uma linha de células da arena é a que está mais a Oeste.

Cada célula da arena pode conter um dos seguintes caracteres:

‘.’ — célula normal;

‘\*’ — célula que contém uma figurinha da Copa;

‘#’ — célula que contém uma pilastra;

‘N’, ‘S’, ‘L’, ‘O’ — célula onde o robô inicia o percurso (única na arena). A letra representa a orientação inicial do robô (Norte, Sul, Leste e Oeste, respectivamente).

A última linha da entrada contém uma sequência de  $S$  caracteres dentre ‘D’, ‘E’ e ‘F’, representando as instruções do robô. O último caso de teste é seguido por uma linha que contém apenas três números zero separados por um espaço em branco.

**Saída:** Para cada caso de teste seu programa deverá imprimir uma linha, contendo o número de bilhetes do pacote que contém outro bilhete com o mesmo número da sequência.

Exemplo de Entrada	Exemplo de Saída
3 3 2	0
***	1
*N*	3
***	
DE	
4 4 5	
...#	
*#O.	
*.*.	
*.#.	
FFEFF	
10 10 20	
.....*	
.....*..	
.....*....	
..*.#.....	
...#N.*..*	
.....*	
.....	
.....	
.....	
.....	
FDFFFFFFEFFFFFFEFD	
0 0 0	