

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)**, Campus Curitiba - Central (Paraná – Brasil)  
 Engenharia Mecatrônica - **Departamento Acadêmico de Eletrônica (DAELN)**  
**4ª Prova – Disciplina: Algoritmos – EL71E – Turma: S15/S16**

**Prof. Jean Marcelo Simão** (S15/S16 Teórico-prático), **Prof. Daniel Rossato** de Oliveira (S15 – Prática em Laboratório), **Prof. Luis Alberto Lucas** (S16 – Prática em Laboratório).

**Nome do Aluno:** \_\_\_\_\_

**Horário de Início:** \_\_\_\_\_ **Horário de Fim:** \_\_\_\_\_

1 Forneça exatamente o que o programa abaixo imprime na tela. **(3,0 pontos)**

<pre>#include&lt;stdio.h&gt;  struct estrutura {     int l;     char c; };  int funcao(struct estrutura est);  int main() {     int l;     struct estrutura minhaEstrutura;     minhaEstrutura.l = 7;     minhaEstrutura.c = '*';     l = funcao(minhaEstrutura);     printf("\n%i", l);     return 0; }</pre>	<pre>int funcao(struct estrutura est) {     int i,j;     for(i=0; i &lt; est.l; i++)     {         for(j=0; j &lt; est.l; j++)         {             if(j &lt;= i)                 printf("%c", est.c);         }         printf("\n");     }     return est.l * est.l; }</pre>
--	---

2 Escreva uma função que receba como parâmetro um vetor **int** de 10 posições e uma outra variável **int** x. Esta função deve retornar a média dos valores do vetor, e **modificar** o valor da variável x para o maior valor contido no vetor. **(2,0 pontos)**

3 Um programa para auxiliar na operação de um robô recebe do usuário o caminho que deve ser percorrido. Este caminho é formado por diversas instruções intermediárias, representadas por duas informações: direção e distância. A direção é dada pela primeira letra de um dos pontos cardeais (**Norte, Sul, Leste, Oeste**) e a distância é dada por um número inteiro entre 1 e 100. Por exemplo, a seguinte entrada:

```
L
30
N
10
O
15
```

significa que o robô andar 30 passos para o Leste, depois 10 passos para o Norte, e por fim 15 passos para o Oeste. Escreva um programa que primeiro leia o número de instruções que o robô receberá (no mínimo 1, no máximo 50), receba as instruções, armazene as mesmas em um vetor de estruturas que representem cada instrução, e por fim utilize a função **void** gravaVetor(**struct** instrucao instrucoes[50], int numInstrucoes) para gravar este vetor em um arquivo. Considere a função gravaVetor já implementada. **(5,0 pontos)**

**A interpretação faz parte do conteúdo da prova! Inclua comentários (se for o caso) para deixar explícitas as decisões em relação às interpretações!**

**Gabarito****Questão 3:**

```
#include<stdio.h>

struct Instrucao
{
    char direcao;
    int nPassos;
};

//le um inteiro validado dentro de um intervalo fechado
int leInteiroFaixa(int min, int max)
{
    int verif, valor;
    do
    {
        printf("\nNo minimo %i, no maximo %i: ", min, max);
        fflush(stdin);
        verif = scanf("%i", &valor);
    } while(valor<min || valor>max || verif!=1);
    return valor;
}

//le um char igual a 'N', 'S', 'L' ou 'O'
char leDirecao()
{
    int verif;
    char direcao;
    do
    {
        printf("\nN, S, L ou O? ");
        fflush(stdin);
        verif = scanf("%c", &direcao);
    } while((direcao!='N' && direcao!='S' && direcao!='L' && direcao!='O') || verif!=1);
    return direcao;
}

int main()
{
    int nInstrucoes, i;
    struct Instrucao caminho[50];

    printf("Quantas instrucoes o robo recebera? ");
    nInstrucoes = leInteiroFaixa(1, 50);

    for(i=0;i<nInstrucoes;i++)
    {
        printf("\n%ia. instrucao:\nQual a direcao? ", i+1);
        caminho[i].direcao = leDirecao();
        printf("\nQual o numero de passos? ");
        caminho[i].nPassos = leInteiroFaixa(1,100);
    }

    gravaVetor(caminho, nInstrucoes);
    return 0;
}
```

**A interpretação faz parte do conteúdo da prova! Inclua comentários  
(se for o caso) para deixar explícitas as decisões em relação às interpretações!**

**Questão 1:**

```
*  
**  
***  
****  
*****  
*****  
*****
```

49

**Questão 2:**

```
float funcao(int vetor[10], int *x)  
{  
    int i, soma=0;  
    *x = vetor[0];  
    for(i=0;i<10;i++)  
    {  
        soma = soma + vetor[i];  
        if(vetor[i] < *x)  
            *x = vetor[i];  
    }  
  
    return soma/10.0;  
}
```