

Lista 7

QUESTÃO 1

```
int i=99, j; //Declara duas variáveis do tipo inteiro: i, j, sendo que o valor de i=99.
int *p; //Declara um ponteiro chamado p.
p = &i; //O ponteiro p recebe o valor do endereço da variável i.
j = *p + 300; //A variável j recebe o valor da soma do conteúdo do ponteiro p mais 300.
```

QUESTÃO 2

i	j	p	&i	&j	*p
99	0	?	100	104	?
99	0	!	100	104	!
99	0	100	100	104	99
99	399	100	100	104	99

QUESTÃO 3

```
int a=5, b=12; // Cria duas variáveis do tipo inteiro e atribui valor a elas, uma variável 'a'
               // com valor 5 e uma variável 'b' com valor 12.
int *p; //Cria um ponteiro 'p'.
int *q; //Cria um ponteiro 'q'.
p = &a; // Atribui ao ponteiro 'p' o valor do endereço da variável 'a'.
q = &b; // Atribui ao ponteiro 'q' o valor do endereço da variável 'b'.
int c = *p + *q; // Cria uma variável 'c' e lhe atribui o valor da soma do conteúdo do ponteiro
               // 'p', que é 5, mais o conteúdo do ponteiro 'q', que é 12, ou seja, a variável c
               // fica com valor de 17.
```

QUESTÃO 4

a	b	c	&a	&b	&c	p	q	*p	*q
5	12	?	100	104	?	?	?	?	?
5	12	?	100	104	?	!	?	!	?
5	12	?	100	104	?	!	!	!	!
5	12	?	100	104	?	100	!	5	!
5	12	?	100	104	?	100	104	5	12
5	12	17	100	104	108	100	104	5	12

QUESTÃO 5

```

int i=7, j=3, c; // Declara três variáveis do tipo inteiro, uma chamada 'i' com valor 7, outra
                // chamada 'j' com valor 3, e outra chamada 'c'.
int *p; // Declara um ponteiro 'p'.
int **r; //Declara um ponteiro de ponteiro 'r'
p = &i; // Atribui ao ponteiro 'p' o valor do endereço da variável 'i'
r = &p; // Atribui ao ponteiro de ponteiro 'r' o valor do endereço do ponteiro 'p'.
c = *r + j; // Atribui a variável 'c' o valor da soma do conteúdo do conteúdo do ponteiro 'r'
            // mais o valor da variável 'j'.

```

QUESTÃO 6

i	j	c	&i	&j	&c	&p	*p	**r	&r
7	3	?	100	104	?	?	?	?	?
7	3	?	100	104	?	108	!	?	?
7	3	?	100	104	?	108	!	!	112
7	3	?	100	104	?	108	100	!	112
7	3	?	100	104	?	108	100	108	112
7	3	10	100	104	116	108	100	108	112

QUESTÃO 7

- a) `p == &i;` //O resultado é 1, que significa true.
b) `*p - *q;` //É a subtração de 3 - 5 que da -2.
c) `*&p;` //É o endereço da variável 'i'
d) `3* - *p/(*q)+7;` //O resultado é 6.

QUESTÃO 8

```

#include <stdio.h>
int main(){
    int i = 3, *ptrl = &i;
    printf("\nValor de i: %d, Valor do ponteiro: %d", i, ptrl);
    *ptrl = 5;
    printf("\nNovo i: %d", i);
    return 0;
}

```

QUESTÃO 9

```

#include <stdio.h>
#include <ctype.h>
void maiusculas(char *str) {
    while (*str) {
        *str = toupper(*str);
        str++;
    }
}

```

```

}
int main() {
    char texto[100];
    printf("Digite o texto: ");
    fgets(texto, sizeof(texto), stdin);
    maiusculas(texto);
    printf("Texto em maiúsculas: %s\n", texto);

    return 0;
}

```

QUESTÃO 10

```

#include <stdio.h>
int main(){
    int vetor[5], i, soma = 0;
    int *ptrV;
    printf("\nDigite o vetor de 5 numeros inteiros: ");
    for(i = 0; i < 5; i++){
        scanf("%i", &vetor[i]);
    }
    ptrV = &vetor;

    for(i = 0; i < 5; i++){
        soma += *(ptrV + i);
    }
    printf("\nA soma de todos os itens e: %i", soma);
    return 0;
}

```

QUESTÃO 11

```

#include <stdio.h>
int main(){
    int vetor[3], *ptrV, i;
    printf("\nDigite 3 valores inteiros: ");
    for(i = 0; i < 3; i++){
        scanf("%d", &vetor[i]);
    }
    ptrV = &vetor;
    printf("\nO vetor e: \n");
    for(i = 0; i < 3; i++){
        printf("[%d] ", *(ptrV + i));
    }
    return 0;
}

```

QUESTÃO 12

```

#include <stdio.h>
struct ALUNO{
    char nome[20];

```

```

    int idade;
    float media;
}aluno;
int main()
{
    struct ALUNO *ptrS = &aluno;
    printf("\nInsira o nome do aluno: ");
    gets(ptrS->nome);
    printf("\nInsira a idade do aluno: ");
    scanf("%d", &ptrS->idade);
    printf("\nInsira a media do aluno: ");
    scanf("%f", &ptrS->media);

    printf("\nNome: %s\nIdade: %d\nMedia: %.1f", ptrS->nome, ptrS->idade, ptrS->media);
    return 0;
}

```

QUESTÃO 13

```

#include <stdio.h>
struct ALUNO{
    char nome[20];
    int idade;
    float nota;
}aluno;
int main()
{
    struct ALUNO *ptrS = &aluno;
    printf("\nInsira o nome do aluno: ");
    gets(ptrS->nome);
    printf("\nInsira a idade do aluno: ");
    scanf("%d", &ptrS->idade);
    printf("\nInsira a media do aluno: ");
    scanf("%f", &ptrS->nota);

    printf("\nNome: %s\nIdade: %d\nMedia: %.1f", ptrS->nome, ptrS->idade, ptrS->nota);
    return 0;
}

```

QUESTÃO 14

```

#include <stdio.h>
#include <string.h>
struct PESSOA{
    char nome[20];
    int idade;
    struct endereco{
        char rua[100];
        int numero;
        char cidade[100];
    };
};

```

```

    }end;
}pessoa;
int main()
{
    struct PESSOA pessoa;
    strcpy(pessoa.nome, "João");
    pessoa.idade = 30;
    strcpy(pessoa.end.rua, "Rua das Flores");
    pessoa.end.numero = 123;
    strcpy(pessoa.end.cidade, "São Paulo");

    struct PESSOA *ptrP = &pessoa;

    printf("\nDados Pessoais\nNome: %s\nIdade: %d", ptrP->nome, ptrP->idade);
    printf("\n\nEndereço\nRua: %s\nNumero: %d\nCidade: %s", ptrP->end.rua,
    ptrP->end.numero, ptrP->end.cidade);
    return 0;
}

```

QUESTÃO 15

```

#include <stdio.h>
#include <math.h>
struct PONTO {
    float x;
    float y;
};
float distanciaAteOrigem(struct PONTO *ponto) {
    return sqrt(ponto->x * ponto->x + ponto->y * ponto->y);
}
int main() {
    struct PONTO p;
    printf("Digite as coordenadas do ponto (x, y): ");
    scanf("%f %f", &p.x, &p.y);

    float distancia = distanciaAteOrigem(&p);
    printf("A distancia do ponto ate a origem e: %.2f\n", distancia);

    return 0;
}

```