

## LISTA DE EXERCÍCIOS

### Ponteiros

**OBS:** Para os exercícios que pedem rastreio, execute-os no **Code Blocks**, utilize o Debugger com breakpoints e acompanhe os valores das variáveis na janela Watches. Faça impressão de tela da janela Watches e do console e inclua na resposta junto com o código fonte.

1. Considere o trecho de programa a seguir:

```
int i = 3, j = 5;
int *p, *q;
p = &i;
q = &j;
```

Determine o valor de cada expressão a seguir:

- |                           |                            |
|---------------------------|----------------------------|
| a) *p;                    | b) *q * i;                 |
| c) p == &i;               | d) q < &j;                 |
| e) (*p > i) && !(j < *q); | f) !(*p == i)    (j < *q); |
| g) *p - *q;               | h) i / *q;                 |
| i) *p * *q;               | j) (float)*q / *p;         |
| k) 5.0 * *q / *p - 2;     | l) *q / *p * 5.0 - 2;      |
| m) *q / (*p - 1.0);       | n) 3 * - *p / *q + 7;      |
| o) ++*q / *p;             | p) *p / (float)++*q;       |
| q) **&p + **&q;           | r) ++*q / *p * ++j;        |
| s) *q / *p * (float)++i;  | t) 2 * *q / --*p * ++i;    |

2. Considere as declarações/atribuições a seguir:

```
int x = 10, *px = &x, **ppx = &px, *ptemp, temp;
float y = 3.2, *py = &y, **ppy = &py, *ptemp, ftemp;
```

Considere ainda que as variáveis ocupam os seguintes endereços de memória:

x	y	px	py	ppx	ppy
28FF1C	28FF10	28FF18	28FF0C	22FF14	28FF08

Determine o valor de cada item a seguir:

- |                    |                   |
|--------------------|-------------------|
| a) x;              | b) y;             |
| c) px;             | d) &y;            |
| e) &px;            | f) *py;           |
| g) *ppx;           | h) py;            |
| i) &x;             | j) pftemp = py++; |
| k) *px--;          | l) **ppy;         |
| m) &ppy;           | n) *&px;          |
| o) temp = **ppx++; | p) ++ppx;         |
| q) &ppx;           | r) --py;          |
| s) --**ppx;        | t) ++*py;         |

3. Identifique as saídas do programa a seguir:

```
#include <stdio.h>
#include <conio.h>

int main()
{
    int valor;
    int *p1;
    float temp;
    float *p2;
    char aux;
    char *nome = "Algoritmos";
    char *p3;
    int idade;
    int vetor[] = {31, 45, 27};
    int *p4;
    int *p5;

    // letra (a)
    valor = 10;
    p1 = &valor;
    *p1 = 20;
    printf("(a) %d \n", valor);

    // letra (b)
    temp = 26.5;
    p2 = &temp;
    *p2 = 29.0;
    printf("(b) %.1f \n", temp);

    // letra (c)
    p3 = &nome[0];
    aux = *p3;
    printf("(c) %c \n", aux);

    // letra (d)
    p3 = &nome[4];
```

```

    aux = *p3;
    printf("(d) %c \n", aux);

    // letra (e)
    p3 = nome;
    printf("(e) %c \n", *p3);

    //letra (f)
    p3 = p3 + 4;
    printf("(f) %c \n", *p3);

    //letra (g)
    p3--;
    printf("(g) %c \n", *p3);

    //letra (h)
    p4 = vetor;
    idade = *p4;
    printf("(h) %d \n", idade);

    //letra (i)
    p5 = p4 + 1;
    idade = *p5;
    printf("(i) %d \n", idade);

    //letra (j)
    p4 = p5 + 1;
    idade = *p4;
    printf("(j) %d \n", idade);

    //letra (k)
    p4 = p4 - 2;
    idade = *p4;
    printf("(k) %d \n", idade);

    //letra (l)
    p5 = &vetor[2] - 1;
    printf("(l) %d \n", *p5);

    //letra (m)
    p5++;
    printf("(m) %d \n", *p5);

    //letra (n)
    printf("(n) %d \n", *(p5 - 1));

    return(0);
}

```

4. Supondo que a variável `i` ocupa o endereço `0028FF0C` na memória, identifique qual é a saída do programa a seguir:

```

main()
{
    int i=5, *p;
    p = &i;
    printf("%p %d %d %d %d\n", p, *p + 2, **&p, 3 *
    *p, **&p + 4);
}

```

5. Execute o programa abaixo no Code Blocks, faça o rastreo (utilizando debugger e breakpoints) e identifique a saída. Apresente impressões de tela do rastreo e da saída.

```

int main(int argc, char *argv[])
{
    int a,b,*p1, *p2;
    a = 4;
    b = 3;
    p1 = &a;
    p2 = p1;
    *p2 = *p1 + 3;
    b = b * (*p1);
    (*p2)++;
    p1 = &b;
    printf("%d %d\n", *p1, *p2);
    printf("%d %d\n", a, b);
}

```

6. Considere o programa a seguir em que os valores lidos para as variáveis  $x$  e  $y$  são 3 e 4, respectivamente. Execute o programa no Code Blocks, faça o rastreo (utilizando debugger e breakpoints) e identifique a saída. Apresente impressões de tela do rastreo e da saída.

```

void func(int *px, int *py)
{
    px = py;
    *py = (*py) * (*px);
    *px = *px + 2;
}

void main(void)
{
    int x, y;
    scanf("%d", &x); //3

```

```
scanf("%d",&y); //4
func(&x,&y);
printf("x = %d, y = %d", x, y);
}
```

7. Considere um cadastro de alunos matriculados em uma disciplina, com as seguintes informações para cada aluno:

- Nome do aluno: com até 80 caracteres
- Número de matrícula: representado por um valor inteiro
- Notas obtidas em três provas, P1, P2, P3: representadas por valores reais

(a) Defina uma estrutura em C, denominada aluno, que tenha os campos apropriados para guardar as informações de um aluno, conforme descrito acima.

(b) Escreva uma função que receba como parâmetro um ponteiro para uma estrutura do tipo definido no item anterior e imprima na tela do computador uma linha com o nome do aluno e outra linha com a média obtida nas três provas. Essa função deve seguir o seguinte protótipo:

```
void imprime (struct aluno* a);
```

(c) Escreva um programa para testar a função.