

Nome: Gustavo Kenji Ando

Data: 22/11/2021

Instruções:

- *Poderá ser realizada em dupla.*
- *Entregar as respostas nesse documento.*
- *Entregar até 01/12*

Prática06 - Ponteiros

1. Explique o código abaixo:

```
main() { // Função Main
    int num1, num2, *p, *q; // Declarando as variáveis que serão usadas como int, sendo o p e
o q com * no começo que indica que serão ponteiros
    num1 = 13; // A variável num1 receberá 13
    num2 = 10; // A variável num2 receberá 10
    p = &num1; // A variável p receberá o endereço de memória do num1
    q = p; // A variável q receberá p, que contém o endereço de memória do num1
    *q = 56; // Altera o valor do endereço que q aponta, no caso o endereço que a variável q
recebeu é de num1. Assim, num1 irá se alterar para 56
    printf("v1:%d\n", num1); // Será mostrado na tela o valor de num1, ou seja, 56
    printf("v2:%d\n", num2); // Será mostrado na tela o valor de num2, ou seja, 10
    printf("p:%p\n", p); // Será mostrado na tela o valor de p, ou seja, o endereço de memória
de num1
    printf("q:%p\n", q); // Será mostrado na tela o valor de q, ou seja, o mesmo valor de p, já
que q recebeu p, que é o endereço de memória de num1
    printf("p:%d\n", *p); // Será mostrado na tela o valor que está sendo apontado por p (*p),
ou seja, 56, já que o endereço recebido por p foi de num1
    printf("q:%d\n", *q); // Será mostrado na tela o valor que está sendo apontado por q (*q),
ou seja, 56, já que *q recebeu 56
}
```

2. Faça um programa que solicite um número (inteiro/positivo) e imprima esse número e o seu endereço de memória (por meio de um ponteiro).

#include <stdio.h>

```
int main()
{
    int n, *p;
    printf("Digite um valor: ");
    scanf("%d", &n);
    p = &n;
    printf("Numero: %d\n", *p);
    printf("Endereco: %p\n", p);
}
```

3. Seja o seguinte trecho de programa:

Nome: Gustavo Kenji Ando

Data: 22/11/2021

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

Qual é o valor da variável resultado do tipo int?

a) resultado=p == &i;

resultado = 1

Observação: Se uma condição seja verdadeira (True) em C, ele retorna 1.

b) resultado= *p - *q

resultado = 3-5

resultado = -2

c) resultado=(12 + *p)/(*q)+7;

resultado = (12 + 3)/(5)+7

resultado = 15/5+7

resultado = 3+7

resultado = 10

4. Mostre na tabela abaixo todos os passos (teste de mesa) e identifique qual será a saída

do programa em C, para os valores lidos (x = 5 e y = 2), suponha que o endereço de memória atribuído para x=0028FF40 e y=0028FF44

Teste de mesa				
	px	py	x	y
A	0028FF40	0028FF44	5	2
B	0028FF40	0028FF40	5	2
C	0028FF40	0028FF40	125	2
D	0028FF40	0028FF40	250	2
E	0028FF40	0028FF40	250	2

```
#include<stdio.h>
#include<conio.h>
void func(int *px, int *py) /*A*/
{
    py = px; /*B*/
    *py = (*py) * (*px)*(*py); /*C*/
    *px = *py+*px /*D*/
}
void main(void)
{
    int x, y;
    scanf("%d",&x); /*5*/
    scanf("%d",&y); /*2*/
    func(&x,&y);
    printf("x = %d, y = %d\n", x, y); /*E*/
}
```

5. Qual serão as saídas do seguinte programa?

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

int main() {
```

Nome: Gustavo Kenji Ando

Data: 22/11/2021

```
int    valor;
int    *p1;
float  temp;
float  *p2;
char   aux;
char   *nome = "Algoritmos";
char   *p3;
int    idade;
int    vetor[3];
int    *p4;
int    *p5;

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

// (a) 20

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

// (b) 29.0

/* (c) */

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

// (c) A

/* (d) */
p3 = &nome[4];
aux = *p3;
printf("(d) %c \n", aux);

// (d) r

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

// (e) A

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

// (f) r

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

Nome: Gustavo Kenji Ando

Data: 22/11/2021

```
// (g) o

/* <h> */

vetor[0] = 31;
vetor[1] = 45;
vetor[2] = 27;

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

// (h) 31

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

// (i) 45

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

// (j) 27

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

// (l) 31

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

// (m) 45

/* (n) */
p5++;

printf("(n) %d \n", *p5);

// (n) 27

return(0);
}
```

Saída:

- (a) 20
- (b) 29.0
- (c) A
- (d) r
- (e) A

Nome: Gustavo Kenji Ando

Data: 22/11/2021

- (f) r
- (g) o
- (h) 31
- (i) 45
- (j) 27
- (l) 31
- (m) 45
- (n) 27

6. Qual é o resultado do seguinte programa?

```
#include <conio.h>
#include <stdio.h>
void main(){
    float vet[5] = {1.1,2.2,3.3,4.4,5.5};
    float *f;
    int i;
    f = vet;
    printf("contador/valor/valor/endereco/endereco");
    for(i = 0 ; i <= 4 ; i++){
        printf("\ni = %d",i);
        printf("    vet[%d] = %.1f",i, vet[i]);
        printf("    *(f + %d) = %.1f",i, *(f+i));
    }
}
```

Saída:

```
contador/valor/valor/endereco/endereco
i = 0    vet[0] = 1.1    *(f + 0) = 1.1
i = 1    vet[1] = 2.2    *(f + 1) = 2.2
i = 2    vet[2] = 3.3    *(f + 2) = 3.3
i = 3    vet[3] = 4.4    *(f + 3) = 4.4
i = 4    vet[4] = 5.5    *(f + 4) = 5.5
```

7. Explique o código abaixo:

```
#include <stdio.h> // Incluindo biblioteca <stdio.h> para uso das funções printf
                   // (mostrar na tela) e scanf (ler conteúdo digitado pelo usuário)
int troca (int *a,int *b); // Criando função troca (chamado no final do programa)

main ( ){ // Função Main
    int num1,num2; // Declarando as variáveis que serão usadas como int
    num1=100; // A variável num1 está recebendo 100
    num2=200; // A variável num2 está recebendo 200'
    troca (&num1,&num2); // Usando a função troca com os parâmetros de num1 (100) e num2
    printf (" \nEles agora valem %d %d\n",num1,num2); // Mostrando na tela os valores de
    num1 e num2 após a função: "Eles agora valem 200 100", sendo os valores de num1 e num2
    respectivamente
    getch(); // Função para o programador não fechar sozinho após realizar todos os seus
    comandos
}

Int troca (int *a,int *b){ // Função com dois parâmetros que são ponteiros
    int temp; // Delcarando como int a variável que será usado para trocar os valores
```

Nome: Gustavo Kenji Ando

Data: 22/11/2021

```
temp=*a; // A variável temp recebe a, ou seja, no caso desse programa, 100, já que o
parâmetro utilizado é o endereço da variável num1
*a=*b; // O ponteiro de a recebe o valor do ponteiro de b
*b=temp; // O ponteiro de b recebe o valor de temp que, por sua vez, recebeu o valor
antigo de a, fazendo a troca dos dois valores
}
```

8. Reescreva o código acima sem utilizar ponteiro. Quais são os resultados da sua execução? Qual a diferença observada com e sem o uso de ponteiros.

```
int main(){
    int num1, num2, troca;
    num1 = 100;
    num2 = 200;
    troca = num1;
    num1 = num2;
    num2 = troca;
    printf ("\nEles agora valem %d %d\n", num1, num2);
    getch();
}
```

R: Dessa forma que eu fiz (sem função) é bem simples de se chegar no resultado esperado. Porém, optei dessa forma pois sem ponteiros foi muito difícil se obter mais de um valor pelo retorno da função, sendo muito mais simples utilizando funções o uso dos ponteiros para este programa.

9. Crie a função chamada Concatena (char *str_destino, char *str_origem) usando ponteiros.

Regras: - Concatena duas strings. A string de origem permanecerá inalterada e será anexada ao fim da string de destino.

Obs: não é permitido utilizar strcat. (Faça o principal também-> O usuário deverá digitar duas strings que serão passadas por parâmetros para função concatena).

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

int concatena(char *str_destino, char *str_origem){
    int tam = strlen(str_destino);
    for(int i = 0; i <= strlen(str_origem); i++){
        *(str_destino+i+tam) = str_origem[i];
    }
    return 0;
}

void main(){
    char s1[100], s2[200], *p, *q;
    printf("Digite uma string: ");
    gets(s1);
    printf("Digite outra string: ");
    gets(s2);
    p = s1;
    q = s2;
    concatena(p, q);
    printf("A string concatenada ficou: %s", p);
}
```