PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática
Algoritmos e Estruturas de Dados 1 (AEDS 1)
Profa.: Rosilane Mota

Lista de Exercícios – Ponteiros

Para cada um dos exercícios a seguir, crie um arquivo .c com o main para realização dos testes. O código deve ser todo comentado com indicação das principais decisões sobre os comandos escolhidos.

1. Escreva um programa em C que leia dois inteiros, armazenando-os em variáveis. O programa deve comparar os endereços das variáveis e exibir o maior deles.

2. Explique cada uma das expressões a seguir, indicando a diferença entre elas:

p++; (*p)++; *(p++);

Qual informação se refere a expressão *(p+10)?

3. Se o endereço de uma variável *valor* foi atribuído a um ponteiro *valorPtr*, quais alternativas são verdadeiras? Justifique sua resposta.

a) valor = = &valorPtr
b) valor = = *valorPtr
c) valorPtr = = &valor
d) valorPtr = = *valor

4. Identifique o erro no programa a seguir, de modo que seja exibido o valor 10 na tela.

```
#include <stdio.h>
int main()
{
   int x, *p, **q;
   p = &x;
   q = &p;
   x = 10;
   printf("\n%d\n", &q);
   return(0);
}
```

5. Escreva um programa em C que declare variáveis para armazenar um valor inteiro, um valor real e um caracter. Deve existir no programa ponteiros associados a cada um deles. O programa deve solicitar novos dados para as variáveis e elas devem ser modificadas usando os respectivos ponteiros. Exiba os endereços e os conteúdos de todas as variáveis e ponteiros antes e após a alteração.

6. Observe os dois programas a seguir, Código 1 e Código 2. Qual. É a diferença entre eles? Qual é o valor impresso para ptr em cada um dos códigos? Porque?

Código 1	**Código 2**

int main()
{
int *ptr, i;
ptr = (int *) malloc(sizeof(int));
*ptr = 10;
for(i=0;i<5;i++){
*ptr=*ptr+1;
}
printf("\nptr: %d", *ptr);
free(ptr);
return 0;
}
	int main()
	{
	int *ptr, i;
	ptr = (int *) malloc(sizeof(int));
	*ptr = 10;
	for(i=0;i<5;i++){
	ptr=ptr+1;
	}
	printf("\nptr: %p", ptr);
	free(ptr);
	return 0;
	}

7. Faça um programa que leia dois valores inteiros e chame uma função que receba estes 2 valores de entrada e retorne o maior valor na primeira variável e o menor valor na segunda variável. Escreva o conteúdo das 2 variáveis na tela no programa principal.

8. Faça um programa que leia três valores inteiros e chame uma função que receba estes 3 valores de entrada e retorne eles ordenados, ou seja, o menor valor na primeira variável, o segundo menor valor na variável do meio, e o maior valor na última variável. A função deve retornar o valor 1 se os três valores forem iguais e 0 se existirem valores diferentes. Exibir os valores ordenados na tela no programa principal.

9. Implemente um procedimento que calcule o comprimento e a área de uma circunferência de raio R. Esse procedimento deve obedecer ao cabeçalho a seguir:

void calcCircunferencia (float R, float *compr, float *area)

A área da superfície e o volume são calculados pelas equações:

$C=2*PI*R$ e $A =PI*R^2$

No programa principal faça a leitura do raio, acione o procedimento e exiba os resultados do comprimento e área calculados por ele.

10. Mostre na tabela a seguir todas as alterações dos conteúdos das variáveis (teste de mesa) e identifique qual será a saída do programa em C para os valores lidos (x = 5 e y = 6).

```
void func(int *px, int *py)
{
    px = py;
    *py = (*py) * (*px);
    *px = *px + 2;
}
int main ()
{
    int x, y;
    scanf("%d",&x);
    scanf("%d",&y);
    func(&x,&y);
    printf("x = %d, y = %d", x, y);
    return 0;
}
```

Teste de mesa			
x	**y**	**px**	**py**