Universidade do Estado do Amazonas Programação de Computadores
Professora: Marcela Pessoa
SISTEMAS DE INFORMAÇÃO UEA-EST 2017.2

Aluno(a): Erik Atilio Silva Rey **Matrícula:** 1715310059

Exercícios de Revisão de Ponteiros

1. O que imprime o programa a seguir? Tente entendê-lo e responder. A seguir, execute-o e comprove o resultado.

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

int main()
{
int t, i, M[3][4];
for (t=0; t<3; ++t)
for (i=0; i<4; ++i)
M[t][i] = (t*4)+i+1;
for (t=0; t<3; ++t)
{
for (i=0; i<4; ++i)
printf ("%3d ", M[t][i]);
printf ("\n");
}

return(0);
}
```

R = O programa imprime uma matriz 3x4 com números de 1 a 12 da esquerda pra direita.

2. Qual o valor de y no final do programa? Tente primeiro descobrir e depois verifique no computador o resultado. A seguir, escreva um /* comentário */ em cada comando de atribuição explicando o que ele faz e o valor da variável à esquerda do '=' após sua execução.

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

int main()
{
int y, *p, x;
y = 0;
p = &y;
x = *p;
```

```
x = 4;
(*p)++;
x;
(*p) += x;
printf ("y = %d\n", y);
return(0);
}
```
R = O valor de y = 5

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

int main()
{
int y, *p, x;
y = 0; /* y recebe o valor de 0 */
p = &y; /* p que é um ponteiro recebe o endereço de y */
x = *p; /* x recebe o ponteiro p */
x = 4; /* x recebe o valor 4 */
(*p)++; /* o ponteiro p agrega 1 em seu conteúdo e y = 1*/
x;
(*p) += x; /* p recebe seu valor mais o valor de x que é 4 ,logo 1 + 4 = 5*/
printf ("y = %d\n", y);
return(0);
}
```

3. Qual o conteúdo final:
a) Das variáveis a, b e c?
R = a = 5 , b = 6 , c = 14

b) Do vetor v?
R = v = { 0, 10, 20, 1030, 40, 1050, 60, 1070, 80, 90}

4. Qual o conteúdo final:
a) Das variáveis a, b e c?
R = a=5 , b=6 e c=2005

b) Do vetor v?
R = v { 0, 10, 20, 30, 40, 45, 60, 70, 75, 90 }

5. Qual o conteúdo final:
a) Das variáveis a, b e c?
R = a=5 , b=1006 e c=7

b) Do vetor v?
R = v{ 0, 5, 20, 30, 40, 25, 60, 70, 80, 90 }

6. Explique o que faz este programa

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

void main()
{
float vet[5] = {1.1, 2.2, 3.3, 4.4, 5.7};
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));
printf(" &vet[%d] = %X",i, &vet[i]);
printf(" (f + %d) = %X\n",i, f+i);
}
return 0;
}
```

R = imprime o valor do contador e os dados do vetor, como conteúdo e endereço.

7. O que é um ponteiro?

R = São variáveis que armazenam o endereço de memória de outras variáveis.

8. Qual é o caractere para definir um ponteiro a um tipo de dado?

R = “ * “ ---> conteúdo , “ & “ -----> endereço.

9. Dado o Programa 2, completar as Tabelas A e B:

R =

tabela A:

p1 = “5”
p2 = “4”
p3 = “5”
p4 = “5”

tabela B:

i = “4”
*p2 = “4”
&j = “1007”
&p_2 = “1053”
**p_3 = “5”
**p_4 = “-4452341”
***p_4 = “5”
*p_1 = “5”
*&p_2 = “1243526”

*p_4 = “-1232457”

10. Explique a diferença entre:
a. p++; **R =** soma 1 ao valor de p
b. (*p)++; **R =** soma 1 ao valor do ponteiro *p
c. *(P++); **R =** aumenta em 1 o índice do ponteiro logo p[1]
11. O que quer dizer *(P+10)?
R = se refere a p[10]

12. Explique o que você entendeu da comparação entre ponteiros.
R = Que é possível fazer operações com ponteiros desde que eles sejam do mesmo tipo.

13. Seja o seguinte trecho de programa:

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

Qual é o valor das seguintes expressões?
a. p = = &i; **R =** “True”
b. *p - *q; **R =** “-2”
c. **&p; **R =** “Endereço de memória de i”
d. 3 - *p/(*q) + 7; **R =** “9.4”

14. Qual será a saída desde programa supondo que i ocupa o endereço de memória 4094?

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

R = 4094 7 5 15 9

15. Se i e j são variáveis inteiras e p e q ponteiros para int, quais das seguintes expressões de atribuições são ilegais?
a. p = &i; **sim**
b. *q = &j; **não**
c. p = &*&i; **não**
d. i = (*&)j; **não**
e. i = *&j; **não**
f. i = *&*&j; **não**
g. q = *p; **não**
h. i = (*p)++ + *q;**sim**

16. Qual o valor de y no final do programa? Tente primeiro descobrir e depois verifique no computador o resultado. A seguir, escreva um /*Comentário*/ em cada comando de

atribuição explicando o que ele faz e o valor da variável a esquerda do ‘=’ (atribuição) após sua execução.

```
int main()
 {
int y, *p, x;
y = 0;
p = &y;
x = *p;
x = 4;
(*p)++;

x--;
(*p) += x;
printf ("y = %d\n", y); }
```

R = y = 4

17. Posso definir um ponteiro para int e apontar para um valor float e o programa funcionária de forma adequada. Isto é verdadeiro ou falso? Justifique.
R = Não pois ponteiros só podem apontar para variáveis do mesmo tipo.

18. Sendo P um ponteiro para inteiro, apontado para o endereço de memória 2000, para qual endereço P aponta após um P++?
R = 2001

19. Quais operações aritméticas podem ser usadas com ponteiro?
R = As 4 operações básicas.

20. Escrever um ponteiro para imprimir, do último para o primeiro, cada um dos elementos do vetor vet[10]={1,2,3,4,5,6,7,8,9,0}.
R =

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

int main(){
int vet[10]={1,2,3,4,5,6,7,8,9,0};
int *p;

p = &vet;

for(int i=9;i>=0;i--){
printf("%d ",*(p+i));
}

return 0;
```

}

21. Escrever um programa para ler uma frase qualquer do teclado e imprimir, esta mesma frase um caractere por vez usando ponteiro.
R =

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

int main(){
char *s;register int t;

    s = malloc(80);

    printf("Digite uma palavra: \n");
    gets(s);

    for(t=0;t<strlen(s);t++){
    putchar(s[t]);
    }
    free(s);
    printf("\n");

return 0;
}
```

22. Completar o programa 15 imprimindo a frase ao contrário e o número de espaços em branco que a frase contém, também usando ponteiro.
R =

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

int main(){
char *s;register int t,cont=0;

    s = malloc(80);

    printf("Digite uma palavra: \n");
    gets(s);

    for(t=strlen(s)-1;t>=0;t--){
    if(s[t] == ' '){
    cont+=1;
```

```
    }
    putchar(s[t]);
    }
    free(s);
    printf("\n número de espaços: %d\n",cont);


return 0;
}
```

23. Inicialize um vetor de inteiros aleatoriamente e percorra usando dois ponteiros: um começando no início do vetor e o outro do final até se encontrarem no meio. OBS: o vetor deve conter um número ímpar de elementos.

R =

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

int main(){
//supor um vetor de tamanho 3
    int vet[3],i,*p,*w;

    for(i=0;i<3;i++){
    scanf("%d",&vet[i]);
    }

    p=&vet;w=&vet;

    for(i=0;i<3;i++){
    printf("%d ",*(p+i));
        if(i==2){
            for(int j=1;j>=0;j--){
            printf("%d ",*(w+j));
            }
        }
    }

return 0;
}
```

24. Inicialize um vetor de inteiros aleatoriamente e percorra o vetor usando ponteiro das seguintes maneiras:

a. Endereço_base + deslocamento (Ex: ponteiro++)

R=

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

```
int main(){
   int vet[3],i,*p;

   for(i=0;i<3;i++){
   scanf("%d",&vet[i]);
   }

   p=&vet;

   for(i=0;i<3;i++){
   printf("%d ",*(p++));
   }

return 0;
}
```

b. O ponteiro como vetor (ponteiro indexado)
R =

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

int main(){
   int vet[3],i,*p;

   for(i=0;i<3;i++){
   scanf("%d",&vet[i]);
   }

   p=&vet;

   for(i=0;i<3;i++){
   printf("%d ",*(p+i));
   }

return 0;
}
```

25. Coloque em ordem o programa abaixo:

```
int main ( ){
 int x, *p;
 *p = x+20;
 p=&x;
x=10;
}
```

Com relação ao programa anterior está correto a sintaxe: p = x; Se não, justifique:

R =
em ordem:

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

int main ( ){
   int x, *p;
   p = &x;
   x=10;
   *p = x+20;
   printf("%d ",x);
return 0;
}
```

em relação a sintaxe p = x:
Está errada, pois p é um ponteiro e o correto a se fazer é p = &x.

26. Qual o resultado de n e pn após a execução do programa abaixo?

```
int main(){
        int n=100;
        int *pn;
        printf(“\n n=%d”, n);
        pn = &n;
        *pn=200;
        printf(“\n n=%d”, n); n=2*(*pn);
        printf(“\n *pn=%d”, *pn);
        printf(“\n n=%d”, n);
return 0;
}
```

R =

```
n= 100
n=200
*pn=400
n=400
```

27. Qual a falha no uso do ponteiro no programa abaixo?

```
int main() {
float x,y;
int *p;
x = 100.25;
p = &x;
y = *p;
printf ("x = %f e y = %f",x,y);
return 0;
}
```

R = O ponteiro *p é do tipo “int”.

28. O que sairá na tela após a execução do programa abaixo:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h> // strlen

int main() {
 char *p = "Mensagem inicial";
 int t;
 printf (p);
 printf ("\n");
 for (t = strlen (p) - 1; t > -1; t--)
 {
 printf ("%c",p [t]);
 }
 return 0;
 }
```

R =

Mensagem Inicial
laicinI megasneM