|  |  |
| --- | --- |
| **Área de Ciências Tecnológicas**  **Algoritmos e Programação II**  **Lista de Exercícios VI – Ponteiros** | Centro Universitário Franciscano_P&B_positivo |

1. Como é realizada a declaração de uma variável do tipo ponteiro na linguagem C? Para que serve um ponteiro?

*Declarada como “tipo \*ponteiro;” -*

1. O endereço de uma variável *x* foi atribuído ao ponteiro *p*. Como pode-se alterar o conteúdo de *x* utilizando o ponteiro *px*?

*p = &x; -> \*p = <valor>*

1. O que indicam as seguintes declarações:

float a; - *Declaração de variável float*

float \*b; - *Declaração de um ponteiro \*b*

float c[10]; - *Declaração de um Vetor (também é ponteiro) c[10]*

float d[3][3]; - *Declaração de uma Matriz (também é ponteiro) d[3][3]*

1. Considere que o endereço de uma variável *a* foi atribuído a um ponteiro *pa*. Quais das seguintes expressões são verdadeiras?
2. a==&pa - *Falso*
3. a==\*pa - ***Verdade***
4. pa==\*a – *Fals*o
5. pa==&a – ***Verdade***
6. Considere a variável inteira *x* e o ponteiro para inteiro *px* que recebeu o endereço de *x*. Quais instruções são corretas para fazer a leitura do valor de *x*?

Int x;

Int \*px;

Px = &x;

1. scanf(“%d”,\*px); - Falso
2. scanf(“%d”, px); - **Verdade**
3. scanf(“%d”, &x); - **Verdade**
4. scanf(“%d”, x); - Falso
5. Ainda sobre o enunciado da questão 5. Escreva uma expressão usando ponteiros que atribua ao *x* o valor de *x* dividido por 10.

Int x;

Int \*px;

px = &x;

x = x/10; ~= x = (\*px)/10;

1. Identifique o erro no trecho de código abaixo, apresentando a solução.

main( ){

int a, \*pa;

**pa = &a; -** Faltava Isto.

\*pa=5;

}

1. Considere o trecho de código:

int \*p;

int i=3;

p = &i;

Quais alternativas são falsas?

1. \*p é igual a 3 - **Verdade**
2. p é igual a 3 - Falso
3. p armazena o endereço de i - **Verdade**
4. ao executar \*p=50, i terá o valor 50 - **Verdade**
5. ao ser alterado o valor de i, \*p será modificado

**Verdade ^**

1. ao ser alterado o valor de i, p será modificado Falso ^
2. Considere o trecho de código:
3. int x, \*px, \*\*pi;
4. float a, \*pa, \*\*pf;
5. Quais atribuições são permitidas?
6. pf=&pa;
7. pi=&x;
8. x=10;
9. \*pa=&a;
10. px=&a;
11. \*pf=&a;
12. pi=&pa;
13. \*\*pf=5.4
14. \*px=2;
15. px=&x;

**Utilizando ponteiros, desenvolva os seguintes algoritmos:**

1. Ler uma string e escreve-la verticalmente na tela.
2. Ler uma string e contar quantos caracteres possui.
3. Ler uma string e inverte-la dentro da mesma string.
4. Escrever um programa que leia uma palavra qualquer e verifique se esta palavra é um palíndromo.

#include<stdio.h>

void main(void){

char p[20], \*p1, \*p2;

int flag=0;

p1=p2=p;

printf("Informe uma palavra: ");

scanf("%s", p);

for( ;\*p2;p2++);

p2--;

for(;p1<p2;p1++,p2--)

if (\*p1!=\*p2) flag=1;

if (flag==1) printf("NAO EH PALINDROMO!!!\n");

else printf("EH PALINDROMO!!!\n");

return;

}

1. Ler duas palavras e compará-las. O programa deve informar se as palavras são iguais, em caso contrário, informar se a primeira é maior do que a segunda, se a segunda é maior do que a primeira ou se são diferentes e tem o mesmo tamanho.

#include<stdio.h>

int main(void){

char pa1[20], pa2[20], \*p1, \*p2;

int t1=0, t2=0;

printf("Informe 1a palavra: ");

scanf("%s", pa1);

printf("Informe 2a palavra: ");

scanf("%s", pa2);

for(p1=pa1,p2=pa2;\*p1==\*p2 && \*p1;p1++,p2++);

printf("\*p1 = %c ::: \*p2 = %c\n", \*p1, \*p2);

if (\*p1==\*p2)

printf("\nAs palavras são iguais\n");

else{

printf("\nAs palavras são diferentes\n");

for(p1=pa1;\*p1;p1++, t1++);

for(p2=pa2;\*p2;p2++, t2++);

if(t1==t2) printf("\nAs palavras tem o mesmo tamanho!\n");

else if(t1>t2) printf("\nA primeira eh maior que a segunda!\n");

else printf("\nA segunda eh maior que a primeira!\n");

}

return 0;

}

1. Ler uma palavra e uma letra qualquer. Mostrar a palavra cortada na primeira posição em que a letra informada for encontrada na palavra.

#include<stdio.h>

int main(void){

char pa1[20], \*p1, letra;

printf("Informe uma letra: ");

scanf("%c", &letra);

printf("Informe 1a palavra: ");

scanf("%s", pa1);

p1 = pa1;

for(;\*p1;p1++)

if(\*p1==letra) break;

\*p1='\0';

printf("\n%s\n", pa1);

return 0;

}

1. Ler um vetor com 10 elementos inteiros e um valor inteiro. Apresentar como resultado o número de vezes que o valor aparece no vetor.

#include<stdio.h>

int main(void){

int vet[10], \*ptr, valor, cont=0;

int i;

ptr=vet;

for(i=0;i<10;i++){

printf("vet[%d] = ", i);

scanf("%d", ptr+i);

}

printf("Informe um valor: ");

scanf("%d", &valor);

for(i=0;i<10;i++, ptr++){

if (\*ptr==valor) cont++;

printf("%d ", \*ptr);

}

printf("\nO valor %d foi encontrado %d vez(es) no vetor\n", valor, cont);

system("pause");

return 0;

}

1. Ler dois vetores com 15 elementos inteiros e um vetor com 15 posições, que armazena as quatro operações aritméticas. Apresentar o resultado de cada operação executada, considerando as posições respectivas nos três vetores.

#include<stdio.h>

#define N 15

int main(void){

int a[15], b[15], \*ptrA, \*ptrB;

char op[15], \*ptrOP;

int i;

ptrA=a;

ptrB=b;

ptrOP=op;

printf("Informe os elementos do 1o vetor:\n");

for(i=0;i<N;i++){

printf("a[%d] = ", i);

scanf("%d", ptrA+i);

}

printf("Informe os elementos do 2o vetor:\n");

for(i=0;i<N;i++){

printf("b[%d] = ", i);

scanf("%d", ptrB+i);

}

printf("Informe as operacoes: +, \*, \, -\n");

for(i=0;i<N;i++){

fflush(stdin);

printf("Operacao[%d] = ", i);

scanf("%c", ptrOP);

ptrOP++;

}

ptrOP=op;

for(i=0;i<N;i++){

if(\*ptrOP=='+') printf("%d + %d = %d\n", \*ptrA, \*ptrB, \*ptrA + \*ptrB);

else if(\*ptrOP=='-') printf("%d - %d = %d\n", \*ptrA, \*ptrB, \*ptrA - \*ptrB);

else if(\*ptrOP=='\*') printf("%d \* %d = %d\n", \*ptrA, \*ptrB, \*ptrA \* \*ptrB);

else if(\*ptrOP=='/') printf("%d / %d = %d\n", \*ptrA, \*ptrB, \*ptrA / \*ptrB);

else printf("Operacao invalida!!!\n");

ptrOP++;

ptrA++;

ptrB++;

}

return 0; }