**Ponteiros do tipo void \***

1 – Encontre os erros existentes no seguinte código e modifique de forma estar em coerência com a linguagem C em relação a ponteiros do tipo void:

#include <stdio.h> /\* printf() \*/

#include <stdlib.h> /\* malloc() \*/

typedef enum{INT, FLOAT, STRING} OPCAO;

/\* Declaração de uma função sem tipo de retorno definido \*/

void \*funcao(OPCAO);

int main(int argc, char \*\*argv)

{

 void \*ponteiro; /\* declaração de uma variável ponteiro \*/

 ponteiro = funcao(INT);

 printf("\nPonteiro inteiro = %d\n", \*ponteiro);

 ponteiro = funcao(FLOAT);

 printf("\nPonteiro float = %f\n", \*ponteiro);

 ponteiro = funcao(STRING);

 printf("\nPonteiro string = %s\n", \*ponteiro);

 return (0);

}

/\* Função que retorna um ponteiro para o valor lido \*/

void \*funcao(OPCAO opcao)

{

 int \*pointer\_inteiro;

 float \*pointer\_float;

 char \*pointer\_string = "frase retornada";

 switch (opcao)

 {

 case INT:

 /\* alocando memoria para um inteiro \*/

 pointer\_inteiro = (int \*) malloc(sizeof(int));

 \*pointer\_inteiro = 747;

 return pointer\_inteiro;

 break;

 case FLOAT:

 /\* alocando memoria para um float \*/

 pointer\_float = (float \*) malloc(sizeof(float));

 \*pointer\_float = 8.76656;

 return pointer\_float;

 break;

 case STRING:

 return pointer\_string;

 break;

 }

 return (NULL);

}

2 – Escreva um programa que receba do usuário três valores do usuário passando como parâmetro um valor que represente um valor qualquer, um valor que represente uma operação de **raiz quadrada** ou **potencia**, e outro valor que represente o tipo de retorno **inteiro** ou **ponto flutuante**. Imprima na saída padrão de acordo com o tipo de retorno escolhido.

3 – Escreva um programa que receba dois valores passados por linha de comando passando como parâmetro para uma função, onde um valor represente um valor qualquer e outro que represente o tipo de retorno **inteiro**, **ponto flutuante**, imprimindo na saída padrão os valores de acordo com o parâmetro de tipo passado por linha de comando.

4 – Escreva um programa que receba um valor passado por linha de comando e passe como parâmetro para uma função que deve identificar se é **inteiro** ou **ponto flutuante**, retornando o valor de acordo com a seguinte enumeração e imprimindo o resultado:

typedef enum{INT, FLOAT} OPCAO;

5 – O código a seguir esta correto? Se estiver qual o resultado? Se não estiver, corrija os erros identificados?

#include <stdio.h>

int main(int argc, char \*argv[])

{

 int valor[2] = {10, 20};

 float valor2[2] = {0.5, 4.65};

 void \*ponteiro;

 ponteiro = &valor;

 printf("%d\n", (int \*)ponteiro);

 printf("%d\n", (int \*)ponteiro+1);

 ponteiro = &valor2;

 printf("%0.2f\n", (float \*)ponteiro);

 return (0);

}

6 – Escreva um programa que possua uma função **troca** que receba como parâmetro dois valores **inteiros** ou **pontos flutuantes**, essa função terá dois ponteiros **void \***, com o seguinte protótipo:

void troca(void \*a, void \*b);

Essa função trocará os valores dos ponteiros independentemente dos tipos existentes nos ponteiros e imprima os valores de cada ponteiro;