

Exercício de Programação Estruturada 4

Exercício de Programação Estruturada 4

1.

O endereço da memória onde a variável está armazenada.

2.

O endereço onde soma está armazenada

3.

&temp corresponde a 16892

&dist corresponde a 16896

&date corresponde a 16900

&miles corresponde a 16908

4.

```
#include <stdio.h>

int main()
{

char key, choice;
int num, count;
float yield, price;

printf("%d\n", &key);
printf("%d\n", &choice);
printf("%d\n", &num);
printf("%d\n", &count);
printf("%d\n", &yield);
printf("%d\n", &price);

return 0;
}
```

5.

Variável: pt_num

Endereço: 500

Valor: 8096

Variável: amt_addr

Endereço: 564

Valor: 16256

Variável: z_addr

Endereço: 8024

Valor: 20492

Variável: num_addr

Endereço: 10132

Valor: 18938

Variável: pt_day

Endereço: 14862

Valor: 20492

Variável: pt_yr

Endereço: 15010

Valor: 694

Variável: years

Endereço: 694

Valor: 1987

Variável: m

Endereço: 8096

Valor:

Variável: amt

Endereço: 16256

Valor: 154

Variável: firstnum

Endereço: 18938

Valor: 154

Variável: balz

Endereço: 20492

Valor: 25

Variável: k

Endereço: 24608

Valor: 154

6.

Não, temp deveria ser declarado como um inteiro e não um ponteiro porque a variável temp não existe ainda, então não podemos achar o endereço de memória dela.

7.

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

int calculateRoot(float* a, float* b, float* c, float* x1, float* x2)
{
    float delta;
    if (*a == 0.0)
    {
        printf("Se a = 0, a equação é de primeiro grau.\n");
        exit(0);
    }
    delta = *b * *b - 4.0 * *a * *c;
    if (delta < 0.0)
    {
        printf("Se Δ < 0, não existe raiz real.\n");
        exit(0);
    }
    else if (delta == 0.0)
    {
        *x1 = -(*b) / 2 * *a;
        *x2 = *x1;
    }
    else
    {
        *x1 = (-(*b) + sqrt(delta)) / (2.0 * *a);
        *x2 = (-(*b) - sqrt(delta)) / (2.0 * *a);
    }
}

int main() {
    float a, b, c, x1, x2;
    printf("Considerando uma equação do 2º grau da forma AX²+BX+C=0;\nInsira
```

```

o valor de A: ");
    scanf("%f", &a);
    printf("Insira o valor de B: ");
    scanf(" %f", &b);
    printf("Insira o valor de C: ");
    scanf(" %f", &c);
    calculateRoot(&a, &b, &c, &x1, &x2);
    printf("As raizes da função são %.2f e %.2f", x1, x2);
return 0;
}

```

9.

```

#include <stdio.h>

void ordenar(float* a, float* b, float* c)
{
    float temp;
    if (*a > *b)
    {
        temp = *a;
        *a = *b;
        *b = temp;
    }
    if (*b > *c)
    {
        temp = *b;
        *b = *c;
        *c = temp;
    }
    if (*a > *b)
    {
        temp = *a;
        *a = *b;
        *b = temp;
    }
}

int main()
{
    float a, b, c;
    printf("Digite A: ");
    scanf("%f", &a);
    printf("Digite B: ");

```

```
scanf(" %f", &b);  
printf("Digite C: ");  
scanf(" %f", &c);  
  
ordenar(&a, &b, &c);  
printf("A = %.2f\nB = %.2f\nC = %.2f\n", a, b, c);  
  
return 0;  
}
```