	Instituto Federal do Triângulo Mineiro - Campus Paracatu
INSTITUTO FEDERAL Triângulo Mineiro Campus Paracatu	**Manutenção e Suporte em Informática**
	Exercícios - Unidade 05
	Estrutura Condicional – Depuração – Código Fonte

| **Disciplina:** Fundamentos de Programação de Computadores | **Turma:** MSI3PA |
| **Professor:** Edwar Saliba Júnior | |

1) Indique a saída dos códigos-fontes a seguir:

1)
```c
int main(){
    float X, Y:

    X = 10;
    Y = X * X;
    X = X * 2;
    X = X * X;
    Y = 2;

    printf("X = %f – Y = %f", X, Y);

    return 0;
}
```

2)
```c
int main(){
    int X, Y;
    float Z;

    X = 10;
    Y = 100;
    Z = 1000;
    Z = Z – (X + Y) * 4;
    X = 5;
    Z = Z / 2 + X;

    printf(("X = %d – Y = %d – Z = %f",
    X, Y, Z);

    return 0;
}
```

3)
```c
int main(){
    float X, Y;

    X = 4 * pow((3 + (15 / (10-5))),2);
    Y = X * X;
    X = X * X;
    Y = Y / X;

    printf("X = %f – Y = %f", X, Y);

    return 0;
}
```

4)
```c
int main(){
    int X, Y;

    X = pow(1,15);
    Y = 27 / (1 + X + (5 / 2 – 1.5));
    X = X * Y;
```

```c
    if((X + 2) > Y)
        printf("X = %d – Y = %d", X * X,
Y);
    else
        printf("X = %d – Y = %d", X, Y *
Y);

    return 0;
}
```

5)
```c
int main(){
    float X, Y;

    X = 17 – pow(2,4);
    Y = X * 3 – 2;
    if(X < Y)
        Y = 4 / 3;
    else {
        X = X * 2;
        if(X > Y)
            printf("X = %f – Y = %f", X *
X, Y + 1);
        else
            printf("X = %f – Y = %f", X +
1, Y * Y);
    }
    return 0;
}
```

6)
```c
int main(){
    float X, Y;

    X = 4;
    Y = 2;
    if((X * X) < Y)
        Y = 9 / 3;
    else {
        if(X == (2 * Y))
            Y = pow(Y,3);

        if(Y != (X * 2))
            X = 12;
            printf("X = %f – Y = %f",X +
1, Y * Y);
        else
            Y = X;
            printf("X = %f – Y = %f",
pow(X,Y), Y * Y);

    }
    return 0;
}
```

2) Indique a saída dos trechos abaixo, considerando:

A = 8;   B = 1;   C = 15;   D = 3;

a)
```
if(D > 5)
   X = (A + B) * D;
else
   X = (A – B) / C;

printf(“X = %f”, X);
```

b)
```
if((A > C) && (B != 7)){
   X = (A + 1) * (B – 2);
   X = X * -1;
}
else{
   X = (A * B) / D * (C + D);
   X = X * -1;
}
printf(“X = %f”, X);
```

c)
```
if((A == 2) || !(B < 7))
   X = (A + 2) * (B – C);
else
   X = (A + B) / D * (C – D);

printf(“X = %f”, X);
```

d)
```
if(!((A > 2) || !(B < 7))){
   X = A + (B – D);
   X = 15 * 2 + 1 + X * -1;
}
else{
   X = A – X * -1;
   X = X * C + 1;
}
printf(“X = %f”, X);
```

e)
```
if(!(A > D) || !(!(B < 7)))
   X = A + B * B;
else
   X = A / pow(B,3);

printf(“X = %f”, X);
```

f)
```
if(!(A > 3) && (B <= (pow(1,3)){
   X = A + D;
   X = X - A + pow(1,125);
}
else {
   X = D / B;
   X = X * C + pow(1,72);
printf(“X = %f”, X);
```

g)
```
if(((C * C) >= 2) && !(B <= A)){
   X = (A + D) / 2 – D;
   X = 4;
}
else{
   X = D * C – 4;
   X = 4;
}
printf(“X = %f”, X);
```

h)
```
if((A >= 2) || (C <= 1)){
   X = (A + D) / 2;
   X = X + D * 2;
}
else{
   X = D * C;
   X = X * C + 1;
}
printf(“X = %f”, X);
```

i)
```
if((A >= D) && !(C != 1)){
   X = (A + D) / 2 – 0.5;
   X = pow(X,3) + 125;
   X = X / (2 * pow(5,2));
   X = ((A + D) / 2 – 0.5) + 5;
   X = X / 2;
}
else{
   X = (A + D) / 2 – 0.5;
   X = X + pow(1,5);
   X = pow(X,3) + 125;
   X = ((X – 216) * 2) / (2 *
pow(5,2));
   X = ((A + D) / 2 – 0.5) + 5;
   X = X / 2;
   X = X + 1;
}
printf(“X = %f”, X);
```

j)
```
if((A >= (C – 2)) || (C <= 1)){
   X = (A + D) / 2;;
   X = pow(A,D) / 2 – 18;
   X = pow(X,D) * 18;
   X = (pow(X,D) - 9) * 18;
   X = (X / 17);
}
else{
   X = D * C;
   X = pow(A,D) / 2 – 18;
   X = pow(X,D) * 18;
   X = (pow(X,D) * 34) * 18;
   X = (X / 9);
}
X = 1;
printf(“X = %f”, X);
```