Universidade Tecnológica Federal do Paraná
Departamento Acadêmico de Informática
CSF13 – Fundamentos de Programação 1
Prof. Bogdan Tomoyuki Nassu
Profa. Leyza Baldo Dorini

(*) 1. Qual será a saída mostrada pelo programa abaixo? (Importante: descubra somente analisando o código, só execute o programa para conferir o resultado.)

```
#include <stdio.h>

int main ()
{
    int foo = 5, bar = 2;
    float foobar = 8.0;

    printf ("%d\n", foo + 3 * bar);
    printf ("%d\n", (foo + 3) * bar);
    printf ("%d\n", (foo * bar) / 8);
    printf ("%f\n", (foo * bar) / foobar);
    printf ("%d\n", foo % 3);
    printf ("%d\n", foo % bar);
    printf ("%d %d\n", foo * -1, -foo);
    printf ("%f\n", foo + 0.1);
    printf ("%d\n", foo / bar * bar);

    return 0;
}
```

Se você está prestando atenção, deve se perguntar agora: os nomes das variáveis não deviam ser significativos? O que significa `foo`, `bar` e `foobar`? Em programação, é normal usar esses nomes em exemplos, eles são como o Fulano, Ciclano e Beltrano da programação. Mas, assim como você não vai chamar o seu filho de Fulano (nem de Foolano), você não vai chamar nenhuma variável real nos seus programas de `foo`! Estes nomes servem somente para exemplos! (Se quiser saber a origem desses nomes, pesquise, ou pergunte para os professores em um horário livre.)

(**) 2. Escreva um programa que calcule e mostre o volume de uma esfera sendo fornecido o valor de seu raio, *r*. A fórmula para calcular o volume é: $(4/3)\cdot\pi\cdot r^3$. Para obter o valor de π, você pode utilizar um valor chamado `M_PI`, que está declarado no arquivo math.h. Por exemplo, para $r = 3$, o volume é 113.097336, e para $r = 15$, o volume é 14137.166941.

(*) 3. Compile o código-fonte abaixo e execute o programa resultante:

```
#include <stdio.h>

int main()
{
    printf ("%f\n", 3.539);
    printf ("%.0f\n", 3.539);
    printf ("%.1f\n", 3.539);
    printf ("%.2f\n", 3.539);
    printf ("%.3f\n", 3.539);
    printf ("%.4f\n", 3.539);
    printf ("%.5f\n", 3.539);

    return 0;
}
```

O que acontece? Você consegue compreender algum padrão que não foi explicado na aula?

(*) 4. Qual será a saída mostrada pelo programa abaixo? (Importante: descubra somente analisando o código, só execute o programa para conferir o resultado.)

```
#include <stdio.h>

int main ()
{
    int foo = 10, bar = 5;

    printf ("%d\n", foo + 10);
    printf ("%d\n", foo + 10);
    printf ("%d\n", foo + 10);

    bar = foo + 1;

    printf ("%d\n", foo);
    printf ("%d\n", bar);
    printf ("%d\n", foo+bar);

    foo = foo + 2;
    printf ("%d\n", foo);

    return 0;
}
```

(**) 5. Escreva um programa em C que recebe 5 números reais, lidos usando a função `scanf`, e calcule e mostre a média dos 2 primeiros números, dos 3 primeiros números, dos 4 primeiros números, e dos 5 números. Por exemplo, se os números forem 1, 2, 3, 4 e 5, as médias serão:

1º e 2º – 1.5
1º a 3º – 2
1º a 4º – 2.5
1º a 5º – 3

Importante: o programa deve usar 5 variáveis para os números, e a linha de código para apresentar todas as médias deve ser sempre:

```
printf ("%f\n", media);
```

, onde `media` é uma única variável que deve ser reaproveitada para todas as médias.

(**) 6. Escreva um programa que converta um dado número de segundos em dias, horas, minutos e segundos. Por exemplo, 7322 segundos correspondem a 0 dia, 2 horas, 2 minutos e 2 segundos.

(*) 7. Todas as linhas do programa abaixo após a declaração das variáveis e antes do `return` possuem um (e somente um) erro no uso da linguagem C (ou seja, não são erros de lógica). Quais são os erros? Dica: para começar, tente identificar os erros mais óbvios apenas analisando o código, mas se tiver dúvidas quanto a alguma estrutura, você pode criar um pequeno programa apenas para testá-la. O erro da última linha NÃO foi mencionado em aula, você vai precisar testar e pesquisar para descobrir o problema!

```
#include <stdio.h>

int main ()
{
    float x1 = 5.0, x2 = 2.0;
    int x3;

    printf ("%f\n", x1 % x2);
    printf ("%f\n", &x1);
    x1 = x2
    printf ("%d\n", x3);
    X2 = 10;
    x1 + 10.0;
    x3 = 039;

    return 0;
}
```

As questões abaixo são opcionais. Faça elas se tiver conseguido chegar até aqui sem dificuldades. Elas envolvem reflexão. Discuta estas questões com os colegas e/ou com os professores!

(****) 8. OPCIONAL! Compile o código-fonte abaixo e execute o programa resultante:

```
#include <stdio.h>

int main()
{
    printf ("%d\n", 999999999);
    printf ("%d\n", 9999999999);
    return 0;
}
```

Ocorreu algum problema? Se sim, diga qual foi o problema e formule uma hipótese sobre o que poderia ter causado o problema. Do contrário, formule uma hipótese sobre o porquê de o código acima poder ser problemático para algumas pessoas mas não para outras. Em ambos os casos, a sua hipótese não precisa ser a explicação correta, o importante é pensar nela!

(****) 9. OPCIONAL! Os programas resultantes dos dois códigos-fonte abaixo produzem a mesma saída:

```
#include <stdio.h>

int main()
{
    printf ("1, 2, 3\n");
    return 0;
}
```

e

```
#include <stdio.h>

int main()
{
    printf ("%d, %d, %d\n", 1, 2, 3);
    return 0;
}
```

Pense e responda com sinceridade: você acredita que os dois códigos produzem o mesmo programa? Se a resposta for “não”, qual poderia ser a diferença entre os programas? (Não importa muito a resposta estar correta, o importante é que você pense nessa questão!)

(****) 10. OPCIONAL! Se usamos variáveis do tipo `int` para inteiros e `float` para valores reais, seria mais inteligente usar somente variáveis do tipo `float`, sempre, já que o conjunto dos números reais contém todos os inteiros. Isso facilitaria a nossa vida e evitaria erros, já que só precisaríamos usar um único tipo de número!

A afirmação acima está ERRADA. Você consegue pensar em uma ou mais razões para usarmos dois tipos diferentes de variáveis (e até mais que isso, como veremos em breve)? Não importa muito a resposta estar correta, o importante é que você pense nessa questão!

(****) 11. OPCIONAL! Se você já aprendeu algo sobre programação, pode ter aprendido o significado do código abaixo:

```
pow (x, 3)
```

Entretanto, na (quase?) totalidade das ocasiões, seria melhor usar simplesmente `x*x*x` (C **não** permite obter o mesmo resultado usando `x^3`, **CUIDADO**!). Por quê? (Se você não sabe o que significa `pow`, não se preocupe, não precisa responder!).