

# Tiny C programs for biologists

(in a not-so-random order)

Workflow :

nano p01.c ► gcc p01.c -lm ► ./a.out

---

[p01.c] Print 'Hello world!!!' to standard output and change line (\n)

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

int main()
{
    printf("Hello world!!!\n");
}
```

---

[p02.c] Declare an integer variable (val), give it a value (100), and type the value of the variable to standard output :

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

int main()
{
    int val;

    val = 100;
    printf("The value is %d\n", val );
}
```

---

[p03.c] Declare a floating point variable (val), give it a value (7.15), and type its value to standard output :

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

int main()
{
    float val;

    val = 7.15;
    printf("The value is %f\n", val );
}
```

---

[p04.c] Declare two variables, give them some values, and type the contents of the variables to standard output :

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

int main()
{
    int    val1;
    float  val2;

    val1 = 100;
    val2 = 7.15;
    printf("The values are %d and %f\n", val1, val2 );
}
```

---

[p05.c] Declare two floats, give them some values, and find and type the maximum of the two variables :

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

int main()
{
    float val1;
    float val2;

    val1 = 3.75;
    val2 = 7.15;
    if ( val1 >= val2 )
    {
        printf("The maximum value is %f\n", val1 );
    }
    else
    {
        printf("The maximum value is %f\n", val2 );
    }
}
```

[p06.c] Instead of assigning values, attempt to read two floats from standard input using scanf(). This assumes that Babis does everything correctly :

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

int main()
{
    float val1;
    float val2;

    scanf("%f %f", &val1, &val2 );
    if ( val1 >= val2 )
    {
        printf("The maximum value is %f\n", val1 );
    }
    else
    {
        printf("The maximum value is %f\n", val2 );
    }
}
```

[p07.c] Same as p06.c but do an extra check whether the two floats are equal (operator == ). Notice also the if() ⇒ else if() ⇒ else construct :

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

int main()
{
    float val1;
    float val2;

    scanf("%f %f", &val1, &val2 );
    if ( val1 == val2 )
    {
        printf("Both values are equal to %f\n", val1 );
    }
    else if ( val1 > val2 )
    {
        printf("The maximum value is %f\n", val1 );
    }
    else
    {
        printf("The maximum value is %f\n", val2 );
    }
}
```

[p08.c] Read a floating point number from standard input and calculate and print its square root. No checks for negative numbers, assume Babis does everything correctly :

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

int main()
{
    float val;
    float y;

    scanf("%f", &val );
    y = sqrt( val );
    printf("The square root of %f is %f\n", val, y );
}
```

[p09.c] The same with p08.c but check for negative numbers :

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

int main()
{
    float val;
    float y;

    scanf("%f", &val );
    if ( val >= 0 )
    {
        y = sqrt( val );
        printf("The square root of %f is %f\n", val, y );
    }
    else
    {
        printf("Negative value detected!\n");
    }
}
```

[p10.c] Same with p09.c but confirm that Babis indeed gave us a number.

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

int main()
{
    float val;
    float y;

    if ( scanf("%f", &val ) == 1 )
    {
        if ( val >= 0 )
        {
            y = sqrt( val );
            printf("The square root of %f is %f\n", val, y );
        }
        else
        {
            printf("Negative value detected!\n");
        }
    }
}
```

[p11.c] Same with p10.c but with different treatment for input errors : If Babis failed to give us a number, then emit an error message and exit the program. Notice that we need an additional `#include <stdlib.h>` to use `exit()`. Notice also the operator for “not equal” (`!=`):

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

int main()
{
    float val;
    float y;

    if ( scanf("%f", &val ) != 1 )
    {
        printf("Error! A number was expected. Abort.\n");
        exit( 1 );
    }

    if ( val >= 0 )
    {
        y = sqrt( val );
        printf("The square root of %f is %f\n", val, y );
    }
    else
    {
        printf("Negative value detected!\n");
    }
}
```