

## Ficha de Trabalho

**Curso:** Programação em C/C++ - fundamentos

**UFCD/Módulo/Temática:** UFCD 0809\_2/N - Programação em C/C++

**Ação:** UFCD 0809\_2/N

**Formador/a:** Hugo Dias

**Data:** 21/01/2025

**Nome do Formando/a:**

## *printf*

1. Implementar o seguinte código C e analisar o *output* final.

```
#include <stdio.h>
int main ()
{
    printf("Inicio:");
    printf("...programa 1.1\n");
    printf("fim.");
    return 0;
}
```

2. O código abaixo exemplifica a utilização do *printf* para escrever tipos de dados no terminal:

```
#include <stdio.h>
int main ()
{
    printf("\n");
    printf("Total de discos SSD(em TB):128\tMemória RAM(GB):32");

    printf("\n\n\n");
    printf("Total de discos SSD(em TB):%d\tMemória RAM(GB):%d ",128,32);
    return 0;
}
```

3. No exemplo anterior foram utilizados valores inteiros, mas também é possível parâmeterizar a função *printf* com outros tipos de dados dados:

```
#include <stdio.h>
int main ()
{
    printf("\n");
    printf("Utilização CPU:%f\n",98.12);
    printf("Arquitectuta CPU:%c\n",'X');
    return 0;
}
```

4. Com a função *puts* é possível a escrita de *strings* e ao contrário do *printf* não permite a escrita de outros tipos de dados:

```
#include <stdio.h>
int main ()
{
    printf("\n");
    printf("Total de discos SSD(em TB):128\tMemória RAM(GB):32\n");
    puts("Total de discos SSD(em TB):128\tMemória RAM(GB):32\n");
    return 0;
}
```

5. Para a escrita de alguns caracteres é necessário a utilização de certos especificadores de formatação:

```
#include <stdio.h>
int main ()
{
    printf("c:\numbers");
    printf("\n");
    printf("c:\\numbers\\terminal%%");
    printf("\n");
    printf("/home/numbers/terminal");
    return 0;
}
```

6. Codificar o código para produzir o seguinte *output*:

```
System conversor
**** 1-Decimal to Binary ****
**** 2-Binary to Decimal ****
**** 3-Decimal to Octal ****
**** 4-Octal to Decimal ****
**** 5-Hexadecimal to Binary ****
**** 6-Binary to Hexadecimal ****
Opção:
```

7. Os especificadores de formatação definem o tipo de dados a imprimir e seguem a seguinte sintaxe:

*%[flags][width][.precision][length]specifier*

No exemplo que se segue são apresentados exemplos que utilizam os especificadores de formatação:

```
#include <stdio.h>
int main ()
{
    printf ("Integers: %d %d \n", -3456, 3456);
    printf ("Characters: %c %c \n", 'z', 80);
    printf ("Long Integers: %ld %ld\n", 4147483647, 5147483647);
    printf ("Some different radices: %d %x %o %#x %ho \n", 100, 100, 100, 100, 100);
    printf ("floats: %4. %E \n", 3.14159, 3.14159);
    printf ("Preceding with empty spaces: %10d \n", 1997);
    printf ("Preceding with zeros: %010d \n", 1997);
    printf ("Width: %*d \n", 15, 140);
    printf ("%s \n", "Educative");
}
```