

Apostila de Arduino
Capítulo 1.

Atividade :

1.Qual a diferença entre os métodos `Serial.print()` e `Serial.println()`?

`_Serial.print()` transmite dados e imprime na mesma linha e `Serial.println()` recebe códigos a mais

e o próximo valor a ser descrito é na linha abaixo.

2.A sentença `char c = n;` faz com que o valor de `n` seja atribuído a `c`, ou seja, a partir desse ponto, as duas variáveis terão exatamente o mesmo valor. Sendo assim, por que essas variáveis são impressas de formas diferentes no monitor serial?

`_`Porque são representadas de formas diferentes internamente, pelo tipo da variável e quantidade de bytes.

3.O programa acima imprime uma tabela que relaciona o código correspondente a cada caractere. Essa tabela é conhecida como a tabela ASCII. Pesquise sobre a tabela ASCII, e verifique:

a. Por que os caracteres de 0 a 31 não são impressos?

Porque são caracteres de controle, para realizar uma ação.

b. Qual o código correspondente ao 'A'?

65

c. Qual o código correspondente ao 'a'?

Qual o código correspondente ao '0' (dígito zero)?

48

d.Qual o código correspondente ao ' ' (espaço)?

32

4.Por que a variável `n` foi declarada como `static`? o que acontece no programa se esse qualificador for omitido? Explique.

- Para indicar que é uma variável permanente. Sem o `static` ele vai virar uma variável local e o valor não vai permanecer, vai ficar se destruindo toda as vezes que o loop reinicia.

5.Por que a contagem de `n` passa automaticamente de 255 para 0? -Porque na tabela ASCII só vai até o 255.

6.Modifique a declaração de `n` para: `static int n = 0;` o que acontece com a contagem?

-Quando chega ao 255 ele continua em ordem crescente, como se tivesse voltado ao 0.

7.Modifique o programa para que ele imprima o valor de `n` nos formatos:

binário, octal, decimal e hexadecimal e depois o caractere `c` (como na tabela ao lado):

8. Para cada caractere enviado ao Arduino, esse programa faz com que o Arduino imprima de volta o código correspondente (e novamente o caractere).

a) Qual o código correspondente ao '{'?

123

b) Quais os códigos correspondentes ao 'a'? e ao 'á'? e ao 'à'?

a: 97 , á: 225 e à: 224

9. Digite uma palavra (seu nome, por exemplo). Observe que o programa processará um caractere de cada vez. Explique.

Na comunicação serial a informação é enviada uma após a outra, como cada letra representa uma informação no programa, cada caractere será entregue uma após a outra.

10. Nesse programa, nenhuma variável foi declarada como static. Por quê?

Porque não tem necessidade de manter seus valores.

11. Leia sobre os Operadores de Comparação, e acrescente a esse programa as seguintes funcionalidades:

a) se o caractere recebido estiver entre '0' e '9', o Arduino deverá imprimir a mensagem: "Categoria:

Dígito numérico";

```
void setup() {  
  Serial.begin(9600); // inicializa comunicação serial (9600 bits/s)  
}  
void loop() {  
  delay (10); // (somente para a simulação)  
  byte n = 0; // declara variável n, e inicializa com valor 0  
  char c; // declara variável c (não inicializa)  
  if ( Serial.available() > 0 ) { // se existem caracteres na fila de entrada  
    c = Serial.read(); // leia 1 caractere da fila de entrada  
    n = c; // atribua a n o valor de c  
    Serial.print ( n ); // imprime o valor como um número  
    Serial.print ( " | " ); // imprime a string " | "  
    Serial.println ( c );  
    if(n >= 0 && n <= 9){  
      Serial.println("Categoria: Dígito um número");  
    } // imprime o valor como um caractere  
  }  
}
```

b) se o caractere recebido estiver entre 'A' e 'Z', o Arduino deverá imprimir a mensagem: "Categoria:

```

Letra maiúscula";
void setup() {
  Serial.begin(9600); // inicializa comunicação serial (9600 bits/s)
}
void loop() {
  delay (10); // (somente para a simulação)
  byte n = 0; // declara variável n, e inicializa com valor 0
  char c; // declara variável c (não inicializa)
  if ( Serial.available() > 0 ) { // se existem caracteres na fila de entrada
    c = Serial.read(); // leia 1 caractere da fila de entrada
    n = c; // atribua a n o valor de c
    Serial.print ( n ); // imprime o valor como um número
    Serial.print ( " | " ); // imprime a string " | "
    Serial.println ( c );
    if(n >= 97&&n <= 122){
      Serial.println("Categoria: Letra maiúscula");
    } // imprime o valor como um caractere
  }
}

```

c) se o caractere recebido estiver entre ‘a’ e ‘z’, o Arduino deverá imprimir a mensagem:

“Categoria:

Letra minúscula”;

```

void setup() {
  Serial.begin(9600); // inicializa comunicação serial (9600 bits/s)
}
void loop() {
  delay (10); // (somente para a simulação)
  byte n = 0; // declara variável n, e inicializa com valor 0
  char c; // declara variável c (não inicializa)
  if ( Serial.available() > 0 ) { // se existem caracteres na fila de entrada
    c = Serial.read(); // leia 1 caractere da fila de entrada
    n = c; // atribua a n o valor de c
    Serial.print ( n ); // imprime o valor como um número
    Serial.print ( " | " ); // imprime a string " | "
    Serial.println ( c );
    if(n >= 97&&n <= 122){
      Serial.println("Categoria: Letra minúscula");
    } // imprime o valor como um caractere
  }
}

```

d) para qualquer outro caractere, imprimir a mensagem “Categoria: outros”

```

void setup() {
  Serial.begin(9600); // inicializa comunicação serial (9600 bits/s)
}
void loop() {
  delay (10); // (somente para a simulação)
  byte n = 0; // declara variável n, e inicializa com valor 0

```

```

char c; // declara variável c (não inicializa)
if ( Serial.available() > 0 ) { // se existem caracteres na fila de entrada
c = Serial.read(); // leia 1 caractere da fila de entrada
n = c; // atribua a n o valor de c
Serial.print ( n ); // imprime o valor como um número
Serial.print ( " | " ); // imprime a string " | "
Serial.println ( c );
if(n != 0 && n != 122){
Serial.println("Categoria: Outros");
} // imprime o valor como um caractere
}
}

```

12. Agora que você já domina a estrutura de controle if / else, volte ao exemplo anterior (Transmitindo dados) e modifique o programa, para que ele imprima somente os caracteres de 0 a 127 (após 127, volta a 0...).

```

void setup() {
Serial.begin(9600); // inicializa comunicação serial (9600 bits/s)
}
void loop() {
delay (10); // (somente para a simulação)
byte n = 0; // declara variável n, e inicializa com valor 0
char c; // declara variável c (não inicializa)
if ( Serial.available() > 0 ) { // se existem caracteres na fila de entrada
c = Serial.read(); // leia 1 caractere da fila de entrada
n = c; // atribua a n o valor de c
Serial.print ( n ); // imprime o valor como um número
Serial.print ( " | " ); // imprime a string " | "
Serial.println ( c );
if(n <= 127){
else( n=0());
}
}
}

```