

## Sistemas Operacionais Embarcados

### Questionário 3

Marcos Breno da Silva Aguiar

150016883

**Para todas as questões, compile-as com o gcc e execute-as via terminal.**

#### 1. Crie um "Olá mundo!" em C

```
...
include <stdio.h>
int main (){
    printf("Olá mundo!\n");
    return 0;
}
...
```

#### 2. Crie um código em C que pergunta ao usuário o seu nome, e imprime no terminal "Ola " e o nome do usuário. Por exemplo, considerando que o código criado recebeu o nome de 'ola\_usuario\_1':

```
```bash
$ ./ola_usuario_1
$ Digite o seu nome: Eu
$ Ola Eu
#include <stdio.h>
int main (){
    char nome[100];
    printf("Digite o seu nome: ");
    scanf("%s", nome);
    printf("Olá %s \n", nome);
    return 0;
}
...

```

#### 3. Apresente os comportamentos do código anterior nos seguintes casos:

##### (a) Se o usuário insere mais de um nome.

```
```bash
$ ./ola_usuario_1
$ Digite o seu nome: Eu Mesmo

Digite o seu nome: marcos breno
Olá marcos ```

```

### 3. Apresente os comportamentos do código anterior nos seguintes casos:

**(a) Se o usuário insere mais de um nome.**

```
```bash
$ ./ola_usuario_1
$ Digite o seu nome: Eu Mesmo

Digite o seu nome: marcos breno
Olá marcos
```
```

**(b) Se o usuário insere mais de um nome entre aspas duplas. Por exemplo:**

```
```bash
$ ./ola_usuario_1
$ Digite o seu nome: "Eu Mesmo"

Digite o seu nome: "marcos Breno"
Olá "marcos"
```
```

**(c) Se é usado um pipe. Por exemplo:**

```
```bash
$ echo Eu | ./ola_usuario_1

echo marcos | ./02
Digite o seu nome: Olá marcos

```
```

**(d) Se é usado um pipe com mais de um nome. Por exemplo:**

```
```bash
$ echo Eu Mesmo | ./ola_usuario_1

echo marcos breno | ./02
Digite o seu nome: Olá marcos
```
```

**(e) Se é usado um pipe com mais de um nome entre aspas duplas. Por exemplo:**

```
```bash
$ echo "Eu Mesmo" | ./ola_usuario_1

echo "marcos breno" | ./02
Digite o seu nome: Olá marcos
marcos@marcos:~/shell-script/lin-c$
```
```

**(f) Se é usado o redirecionamento de arquivo. Por exemplo:**

```
```bash
$ echo Ola mundo cruel! > ola.txt
$ ./ola_usuario_1 < ola.txt

echo marcos breno > nome.txt
```
```

```
./02 < nome.txt  
Digite o seu nome: Olá marcos
```

...

**4. Crie um código em C que recebe o nome do usuário como um argumento de entrada (usando as variáveis `argc` e `*argv[]`), e imprime no terminal "Ola " e o nome do usuário. Por exemplo, considerando que o código criado recebeu o nome de 'ola\_usuario\_2':**

```
```bash  
$ ./ola_usuario_2 Eu  
$ Ola Eu
```

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

```
int main (int argc, char *argv[]){  
    int i = 1;  
    printf("Olá ");  
    while (i < argc) {  
        printf("%s", argv[i]);  
        i +=1;  
    }  
    printf("\n");  
    return 0;  
}
```

...

**5. Apresente os comportamentos do código anterior nos seguintes casos:**

**(a) Se o usuário insere mais de um nome.**

```
```bash  
$ ./ola_usuario_2 Eu Mesmo
```

```
./03 marcos breno  
Olá marcosbreno
```

...

**(b) Se o usuário insere mais de um nome entre aspas duplas. Por exemplo:**

```
```bash  
$ ./ola_usuario_2 "Eu Mesmo"
```

```
./03 "marcos breno"  
Olá marcos breno
```

...

**(c) Se é usado um pipe. Por exemplo:**

```
```bash
$ echo Eu | ./ola_usuario_2
```

```
echo marcos | ./03
Olá
```

```
```
```

**(d) Se é usado um pipe com mais de um nome. Por exemplo:**

```
```bash
$ echo Eu Mesmo | ./ola_usuario_2
```

```
echo marcos breno | ./03
Olá
```

```
```
```

**(e) Se é usado um pipe com mais de um nome entre aspas duplas. Por exemplo:**

```
```bash
$ echo Eu Mesmo | ./ola_usuario_2
```

```
echo "marcos breno" | ./03
Olá
```

```
```
```

**(f) Se é usado o redirecionamento de arquivo. Por exemplo:**

```
```bash
$ echo Ola mundo cruel! > ola.txt
$ ./ola_usuario_2 < ola.txt
```

```
./03 < nome.txt
Olá
```

```
```
```

**6. Crie um código em C que faz o mesmo que o código da questão 4, e em seguida imprime no terminal quantos valores de entrada foram fornecidos pelo usuário. Por exemplo, considerando que o código criado recebeu o nome de 'ola\_usuario\_3':**

```
```bash
$ ./ola_usuario_3 Eu
$ Ola Eu
$ Numero de entradas = 2
```

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

```
int main (int argc, char *argv[]){
```

```

    int i = 1;
    printf("Olá ");
    while (i < argc) {
        printf("%s", argv[i]);
        i +=1;
    }
    printf("\n");
    printf("Número de entradas = %d\n", i);
    return 0;
}

...

```

**7. Crie um código em C que imprime todos os argumentos de entrada fornecidos pelo usuário. Por exemplo, considerando que o código criado recebeu o nome de 'ola\_argumentos':**

```

```bash
$ ./ola_argumentos Eu Mesmo e Minha Pessoa
$ Argumentos: Eu Mesmo e Minha Pessoa

```

```

#include <stdio.h>

int main (int argc, char *argv[]){
    int i = 1;
    printf("Olá ");
    while (i < argc) {
        printf("%s", argv[i]);
        i +=1;
    }
    printf("\n");
    return 0;
}

...

```

**8. Crie uma função que retorna a quantidade de caracteres em uma string, usando o seguinte protótipo:**

**`int Num\_Caracs(char \*string);` Salve-a em um arquivo separado chamado 'num\_caracs.c'. Salve o protótipo em um arquivo chamado 'num\_caracs.h'. Compile 'num\_caracs.c' para gerar o objeto 'num\_caracs.o'.**

```

...

Função num_caracs.h:
"

    #ifndef NUM_CARACS_H
    #define NUM_CARACS_H

    int Num_Caracs(char \*string);

    #endif

```

"

Função num\_caracs.c:

"

```
#include <stdio.h>
#include <stdlib.h>
#include "num_caracs.h"

int Num_Caracs(char \*string){
    int i = 0;
    while(string[i] != '\0'){
        i++;
    }
    return i;
}

int main(){
    int num_char;
    char string[100];
    printf("Digite uma string: ");
    scanf("%s", string);
    num_char = Num_Caracs(string);
    printf("Número de caracteres: %d\n", num_char);
    return 0;
}
...
```

**9. Re-utilize o objeto criado na questão 8 para criar um código que imprime cada argumento de entrada e a quantidade de caracteres de cada um desses argumentos. Por exemplo, considerando que o código criado recebeu o nome de 'ola\_num\_caracs\_1':**

```
```bash
$ ./ola_num_caracs_1 Eu Mesmo
$ Argumento: ./ola_num_caracs_1 / Numero de caracteres: 18
$ Argumento: Eu / Numero de caracteres: 2
$ Argumento: Mesmo / Numero de caracteres: 5
```

```
#include <stdio.h>
#include <stdlib.h>
#include "num_caracs.h"

int Num_Caracs(char \*string){
    int i = 0;
    while(string[i] != '\0'){
        i++;
    }
    return i;
}
```

```

int main(int argc, char \*argv[]){
    int i = 0;
    printf("Olá ");
    while(i < argc){
        printf("Argumento: %s / Numero de caracteres: %d", argv[i],
Num_Caracs(argv[i]));
        i += 1;
    }
    printf("\n");
    return 0;
}
...

```

# 10. Crie um Makefile para a questão anterior.

```

```num_caracs:  num_caracs.h q.c
gcc ./q.c -o out
...

```

# 11. Re-utilize o objeto criado na questão 8 para criar um código que imprime o total de caracteres nos argumentos de entrada. Por exemplo, considerando que o código criado recebeu o nome de 'ola\_num\_caracs\_2':

```

```bash
$ ./ola_num_caracs_2 Eu Mesmo
$ Total de caracteres de entrada: 25

```

```

#include <stdio.h>
#include <stdlib.h>
#include "num_caracs.h"

```

```

int Num_Caracs(char \*string){
    int i = 0;
    while(string[i] != '\0'){
        i++;
    }
    return i;
}

```

```

int main(int argc, char \*argv[]){
    int i = 0; int total = 0; int numero;
    while(i < argc){
        numero = Num_Caracs(argv[i]);
        printf("Argumento: %s / Numero de caracteres: %d\n", argv[i], numero);
        i += 1;
        total += numero;
    }
    printf("Total de caracteres de entrada: %d\n", total);
    return 0;
}

```

```

**12. Crie um Makefile para a questão anterior.**

```
```num_caracs:  num_caracs.h q.c  
gcc ./q.c -o out ```
```