

Disciplina: Sistemas Operacionais Embarcados **Código:** 120961 **Turma:** A

Professor: Diogo Caetano Garcia

Aluno/Matrícula: Fábio Barbosa Pinto – 11/0116356

Questionário: 03_GCC

1. Crie um "Olá mundo!" em C.

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

```
int main(void)
{
    printf("Ola 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':

```
$ ./ola_usuario_1
$ Digite o seu nome: Eu
$ Ola Eu
```

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

```
char name[20];
```

```
int main(void)
{
    printf("Digite seu name: ");
    scanf("%s", name);
    printf("Olá %s \n", name);
    return 0;
}
```

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

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

```
$ ./ola_usuario_1
$ ./ola_usuario_1 Eu Mesmo
```

```
$ ./ola_usuario
$ Digite seu nome: Fabio Barbosa
$ Ola Fabio
```

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

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

```
$ ./ola_usuario  
$ Digite seu nome: "Fabio Barbosa"  
$ Olá "Fabio"
```

(c) Se é usado um pipe. Por exemplo:

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

```
echo Oi | ./ola_usuario  
Digite seu nome: Olá Oi
```

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

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

```
echo Oi Olá | ./ola_usuario  
Digite seu nome: Olá Oi
```

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

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

```
echo "Oi Olá" | ./ola_usuario  
Digite seu nome: Olá Oi
```

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

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

```
echo Ola mundo cruel! > ola.txt  
./ola_usuario < ola.txt  
Digite seu nome: Olá Ola
```

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':

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

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

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

terminal:

```
./ola_usuario_2 Fabio Barbosa
Ola Fabio Barbosa
```

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

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

```
$ ./ola_usuario_2 Eu Mesmo
Ola Eu Mesmo
```

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

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

(c) Se é usado um pipe. Por exemplo:

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

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

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

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

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

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

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

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':

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

```
Codigo:#include <stdio.h>
#include <string.h>

int main(int argc, char const *argv[])
{
    int i;
    printf("Ola");
    for (i=1; i<argc; i++){
        printf(" %s", argv[i]);
    }
    printf("\nNumero de entradas = %d \n", argc);
    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':

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

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

int main(int argc, char const *argv[])
{
    int i;
    printf("Argumentos: ");
    for (i=1; i<argc; i++){
        printf(" %s", argv[i]);
    }
    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'.

```
int Num_Caracs(char *string)
{
    return strlen(string);
}
$ gcc -c num_caracs.c
```

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':

```
$ ./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
```

```
$ gcc -c ola_num_caracs_1.c
$ gcc -o ola_num_caracs_1 ola_num_caracs_1.o num_caracs.o
```

10) Crie um Makefile para a questão anterior.

```
ola_num_caracs: ola_num_caracs_1.o num_caracs.o
    gcc $(CFLAGS) -o ola_num_caracs ola_num_caracs_1.o num_caracs.o
ola_num_caracs_1.o: num_caracs.h
    gcc $(CFLAGS) -c ola_num_caracs_1.c
num_caracs.o: num_caracs.c num_caracs.h
    gcc $(CFLAGS) -c num_caracs.c
clean:
    rm *.o
remove:
    rm ola_num_caracs
```

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':

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

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

```
int main(int argc, char const *argv[])
{
    int i, lenTotal = 0;
    printf("Argumentos: ");
    for(i = 0; i < argc-1; i++)
        lenTotal += Num_Caracs(argv[i]);
```

```
printf("Total de caracteres de entrada: %d \n",lenTotal);  
return 0;  
}
```

12) Crie um Makefile para a questão anterior.

```
ola_num_caracs: ola_num_caracs_2.o num_caracs.o  
    gcc $(CFLAGS) -o ola_num_caracs ola_num_caracs_2.o num_caracs.o  
ola_num_caracs_2.o: num_caracs.h  
    gcc $(CFLAGS) -c ola_num_caracs_2.c  
num_caracs.o: num_caracs.c num_caracs.h  
    gcc $(CFLAGS) -c num_caracs.c  
clean:  
    rm *.o  
remove:  
    rm ola_num_caracs
```