

# IF61C - Fundamentos de Programação

## Passagem de argumentos para a função `main()`

Dainf - UTFPR

Profa. Leyza Baldo Dorini

# A função `main()`

- A função `main()` é uma função especial, que possui um tipo fixo (`int`) e é invocada automaticamente pelo sistema operacional quando este inicia a execução do programa.
- Quando utilizado, o comando `return` informa ao sistema operacional se o programa funcionou corretamente ou não.
- Hoje, iremos passar argumentos de entrada para a função `main()`.

# Argumentos na função main()

A declaração mais completa para a função main() é dada por `int main (int argc, char *argv[]);`

- `argc` (*argument count*): é um inteiro e representa o número de argumentos passados na linha de comando. O nome do programa é considerado nessa contagem. Exemplo:  
`nomeExecutavel parametro1 parametro2` tem `argc` igual a 3.
- `argv` é uma matriz de ponteiros para strings, sendo que `argv[0]` armazena o nome do programa. Nas demais posições, são armazenados ponteiros que apontam para cada um dos argumentos passados para a linha de comando.

# Argumentos na função main()

O programa

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

    printf("Foram passados %d argumentos na linha de comando.\n",argc);
    for(i=0; i < argc; i++)
        printf("%do argumento: %s\n", i, argv[i]);

    return 0;
}
```

tem como saída

```
C:\Users\Leyza> ./teste parametro1 parametro2
Foram passados 3 argumentos na linha de comando.
0o argumento: ./teste
1o argumento: parametro1
2o argumento: parametro2
```

# Atenção!!

Importante: como todos os parâmetros são recebidos como strings, ao receber como parâmetro um valor que precisa ser interpretado como um inteiro, é preciso fazer a conversão!! Exemplo:

```
int main (int argc, char *argv[]) {  
  
    int i;  
  
    printf("Foram passados %d argumentos na linha de comando.\n"  
    for(i=0; i < argc; i++)  
        printf("%do argumento: %s\n", i, argv[i]);  
  
    printf("%d", argv[1]+argv[2]);  
  
    return 1;  
}
```

**Problema:** error: invalid operands to binary + (have 'char \*' and 'char \*')

# Atenção!!

Uso da função atoi()

```
int main (int argc, char *argv[]) {  
  
    int i;  
  
    printf("Foram passados %d argumentos na linha de comando.\n",argc);  
    for(i=0; i < argc; i++)  
        printf("%do argumento: %s\n", i, argv[i]);  
  
    printf("%d", atoi(argv[1])+atoi(argv[2]));  
  
    return 1;  
}
```

Foram passados 3 argumentos na linha de comando.

0o argumento: teste

1o argumento: 10

2o argumento: 20

30

# Passando parâmetros para a função `main()`

Ok, mas como passar parâmetros/argumentos para a função `main()`? Hoje, veremos duas formas:

- Diretamente via linha de comando
- Criando um projeto no CodeBlocks.

# 1: via linha de comando

Como vimos em sala, é possível compilar e executar o programa diretamente via linha de comando. Neste caso, ao executar o programa, é possível passar os argumentos de entrada:

```
C:\Users\Leyza> gcc teste.c -o teste
```

```
C:\Users\Leyza> teste parametro1 parametro2
```

No Linux:

```
$ gcc teste.c -o teste
```

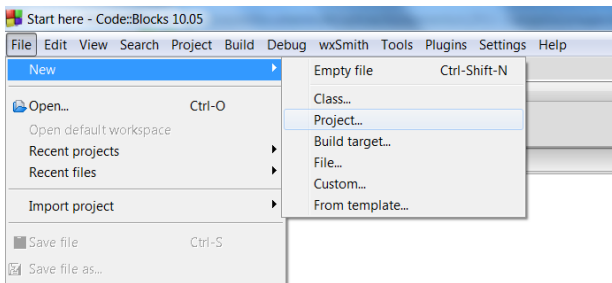
```
$ ./teste parametro1 parametro2
```



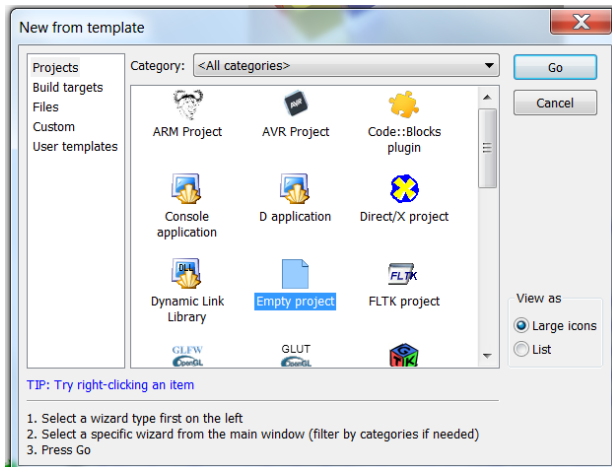
## 2: criando um projeto no CodeBlocks

Neste caso, além de criar um projeto e anexar arquivos .c a ele, é preciso setar os argumentos de entrada! Veja o passo-a-passo na sequência de figuras a seguir.

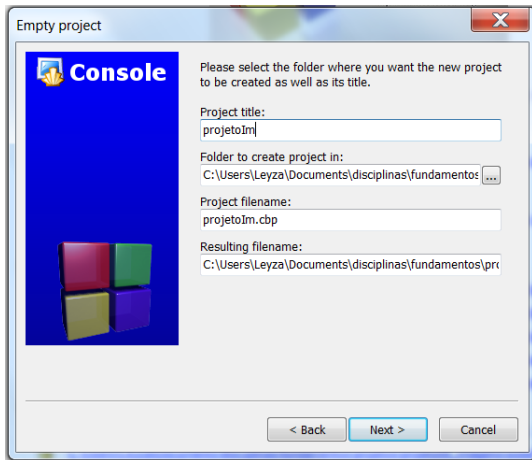
Inicialmente, é preciso criar um projeto



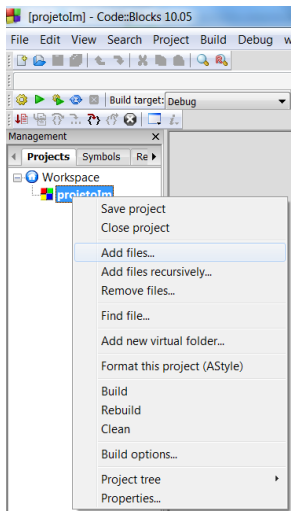
## Escolha a opção Empty Project



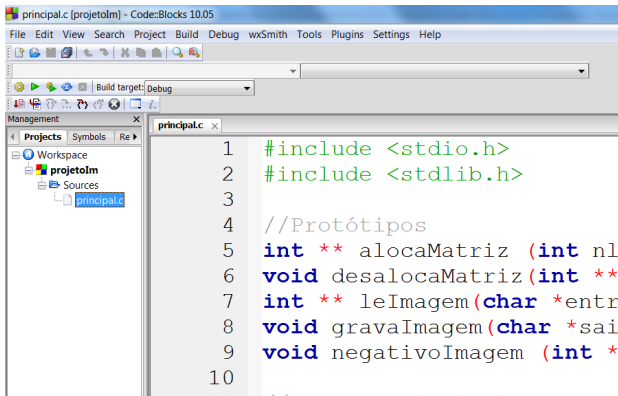
Dê um título ao seu projeto e escolha em qual pasta ele ficará localizado



Agora, é preciso adicionar arquivos .c ao projeto. Para tal, basta clicar com o botão direito do mouse no nome do projeto localizado no canto esquerdo e escolher a opção Add files...

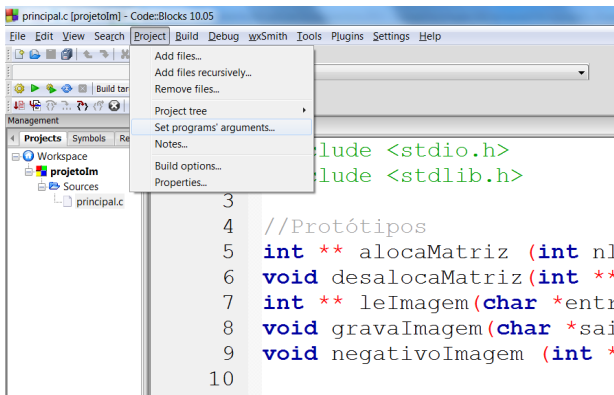


Ao clicar duas vezes sobre o nome do arquivo .c, ele é aberto

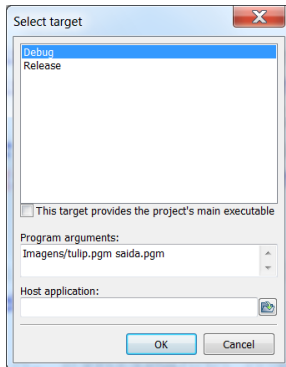


```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  //Protótipos
5  int ** alocaMatriz (int n1
6  void desalocaMatriz(int **
7  int ** leImagem(char *entr
8  void gravaImagem(char *sai
9  void negativoImagem (int *
10
```

Para setar os argumentos de entrada, vá ao menu Project -> Set program's arguments



Na subjanela Program arguments, digite os parâmetros de entrada



No caso deste exemplo, o primeiro argumento é a imagem tulip.pgm, localizada na pasta Imagens. O segundo argumento é a string saida.pgm.



É preciso observar que as duas formas propostas possuem limitações:

- compilar via linha de comando pode ser um tanto trabalhoso se existem vários arquivos .c envolvidos
- a criação de um projeto no CodeBlocks exige que esta IDE seja utilizada

Para evitar estes problemas, nas próximas aulas veremos como utilizar o Makefile para tal. Por enquanto, leia o material de apoio disponibilizado no moodle.