****

/*Файлы Test.cpp и Dockerfile должны быть в одном каталоге

0. Запустить Docker, если еще не запущен.

1. Используя оболочку командной строки cmd, PowerShell или GitBash перейти в каталог с исходными файлами.

2. Построить образ:

docker build -t hello_world . (точка . - говорит, что Dockerfile находится в текущем каталоге; -t задает имя образа)

3. Запустить контейнер:

docker run hello_world (в оболочке командной строки появится соответствующее приветствие)

В таком варианте контейнер docker прекращает работу по завершению выполнения программы.

Для того, чтобы оставить его работать резидентно построение образа нужно модифицировать, добавив

ключ -i:

docker build -it hello_world .

*/

****

```
#****Первый вариант Dockerfile*****************************

# Get the GCC preinstalled image from Docker Hub
FROM gcc:4.9

# Copy the current folder which contains C++ source code to the
Docker image under /usr/src
COPY . /usr/src/dockertest1

# Specify the working directory
WORKDIR /usr/src/dockertest1

# Use GCC to compile the Test.cpp source file
RUN g++ -o Test Test.cpp

# Run the program output from the previous step
CMD ["./Test"]
#*********************************************************

#*****************Второй вариант Dockerfile*******************
# На образе материнской ОС
FROM ubuntu:latest

#Установка gcc с очисткой кэша
 RUN apt update && apt install -y build-essential && apt clean
```

```
    # Copy the current folder which contains C++ source code to the
Docker image under /usr/src
    COPY . /usr/src/dockertest1

    # Specify the working directory
    WORKDIR /usr/src/dockertest1

    # Use GCC to compile the Test.cpp source file
    RUN g++ -o Test Test.cpp

    # Run the program output from the previous step
    CMD ["./Test"]
    #*************************************************************

    //************Файл исходного кода Test.cpp********************


    #include <iostream>

    int main()
    {
        std::cout << "Hello, WORLD!\n";
        std::cin.get();
        return 0;
    }

    //*************************************************************
```